oving Reliability In A Stochastic Communication Netw

THESIS

William Lee Gaught
Captain, USAF

AFIT/GSO/ENS/90D-10

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

AFIT/GSO/ENS/90D-10

Improving Reliability In A Stochastic Communication Network

THESIS

William Lee Gaught
Captain, USAF

AFIT/GSO/ENS/90D-10

Approved for public release; distribution unlimited

AFIT/GSO/ENS/90D-10

Improving Reliability In A Stochastic Communication Network

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science (Space Operations)

William Lee Gaught, B.S., M.S.
Captain, USAF

December, 1990

Preface

This thesis culminates a research project that investigated the behavior of throughput in stochastic communication networks. The experience I gained from this project will be invaluable throughout my Air Force career. I would like to express my deepest appreciation to my thesis advisor, Dr. Yupo Chan. His technical guidance was invaluable. I would also like to thank my reader, Colonel Thomas Schuppe, for his advice and suggestions. A special thanks goes out to Captain Eugene Yim for the effort he put into the programming of the *Formula* program. But most of all I want to thank my wife, Lynn, for her unwaivering support through the past 18 months. It is to these individuals that I dedicate this research.

William Lee Gaught

Table of Contents

Page

Preface ii

Table of Contents iii

List of Figures xii

List of Tables xiii

Abstract xv

I. Introduction 1

1.1 Background 1

1.2 Research Objectives 2

1.3 Assumptions 3

II. Literature Review 4

2.1 Notation 4

2.2 Network Representation 5

2.2.1 Graph Theory. 5

2.2.2 Matrices. 6

2.2.3 Component Reliability and Capacity 6

2.3 Flow Models 8

2.4 Network Reliability 9

2.4.1 Binary Components. 9

2.4.2 Multimode Components. 13

2.4.3 Dependent Failures. 14

Page

2.5 Lower and Upper Bounds of Expected Maximum Flow . . 14
2.5.1 Lower Bound of Expected Maximum Flow. 14
2.5.2 Upper Bound of Expected Maximum Flow. 15
2.6 Formula 16
2.7 Simulation 18
2.8 Summary 19

III. Formulations of Models 21
3.1 Assumptions and Restrictions 22
3.2 Decision Variables 22
3.3 Formulation of Investment Strategy Model 1 23
3.3.1 Objective Function. 23
3.3.2 Constraints. 24
3.3.3 Model. 24
3.4 Formulation of Investment Strategy Model 2 25
3.4.1 Objective Function. 25
3.4.2 Constraints. 25
3.4.3 Model. 26
3.5 Formulation of Investment Strategy Model 3 26
3.5.1 Objective Function. 26
3.5.2 Constraints. 27
3.5.3 Model. 27
3.6 Formulation of Investment Strategy Model 4 28
3.6.1 Objective Function. 28
3.6.2 Constraints. 28
3.6.3 Model. 29
3.7 Formulation of Investment Strategy Model 5 30
3.7.1 Objective Function. 30

Page

3.7.2 Constraints. 30

3.7.3 Model. 31

3.8 Formulation of Investment Strategy Model 6 31

3.8.1 Objective Function. 32

3.8.2 Constraints. 32

3.8.3 Model. 33

3.9 Formulation of Investment Strategy Model 7 33

3.9.1 Objective Function. 33

3.9.2 Constraints. 34

3.9.3 Model. 34

3.10 Nonlinear Costs of Capacity and Reliability 35

3.11 Summary 36

IV. Methodology 37

4.1 Understanding the Problem 37

4.2 Model Formulation 38

4.3 Enhancements to Formula 39

4.4 Design of Experiment 40

4.5 Analysis 41

4.6 Use of Network Packages 42

4.7 Application of Models in a Space Environment 42

4.8 Summary 42

V. Formula Version 2.0 44

5.1 Features of Formula Version 1.0 44

5.2 Enhancements to Version 1.0 45

5.3 Summary 45

		Page
VI.	Sample Networks	46
6.1	Network 1	46
6.2	Network 2	47
6.3	Network 3	48
6.4	Network 4	49
6.5	Network A	50
6.6	Summary	50
VII.	Design of Experiment	55
7.1	Assumptions	55
7.2	Tools	56
7.2.1	Arity/Prolog Interpreter.	56
7.2.2	Formula Version 2.0.	56
7.2.3	Maxflo.	57
7.2.4	LP/MIP-83.	57
7.2.5	GINO.	57
7.2.6	Computer Equipment.	58
7.3	Sample Networks	58
7.4	Files.	58
7.5	Actual Experiment	59
7.5.1	Convert Network.	61
7.5.2	Calculate Original Parameters for the Network.	61
7.5.3	Determine the Optimum Investment Strategies.	62
7.5.4	Implement the Optimum Investment Strategies.	64
7.5.5	Calculate Final Parameters for the Network.	65
7.5.6	Compare Final Parameter Values.	67
7.5.7	Sensitivity Analysis.	67
7.6	Summary	68

		Page
VIII.	Results and Analysis	69
	8.1 Analysis of Network 1	70
	8.2 Analysis of Network 2	74
	8.3 Analysis of Network 3	76
	8.4 Analysis of Network 4	79
	8.5 Analysis of Network A	83
	8.6 Lower Bound Versus Budget	88
	8.7 Summary	90
IX.	Solving Models Using Network Packages	98
	9.1 Decision Variables	98
	9.2 Solving Lower and Upper Bound Models	99
	9.2.1 Lower Bound Solution Using Netflow.	100
	9.2.2 Solving Upper Bound Using Either Netflow or Maxflo.	101
	9.3 Solving Investment Strategy Model 1 Using Netflow	103
	9.4 Solving Investment Strategy Model 4 Using Netflow	105
	9.5 Comparing Investment Strategy Model 3 with Investment Strategy Model 4	106
	9.5.1 Investment Strategies for Network 2.	107
	9.5.2 Investment Strategies for Network 3.	109
	9.5.3 Investment Strategies for Network 4.	109
	9.6 Summary	109
X.	Conclusions and Recommendations	112
	10.1 Conclusions	112
	10.2 Recommendations	114
	10.3 Summary	115

		Page
Appendix A.	Reports for Sample Networks	116
A.1	Network 1	116
A.1.1	Formula Input File	116
A.1.2	Paths	117
A.1.3	Lower Bound For Original Network	118
A.1.4	Upper Bound For Original Network	120
A.1.5	Investment Strategy Model 1	122
A.1.6	Investment Strategy Model 3	124
A.1.7	Output From Maxflo	125
A.2	Network 2	126
A.2.1	Formula Input File	126
A.2.2	Paths	127
A.2.3	Lower Bound For Original Network	128
A.2.4	Upper Bound For Original Network	130
A.2.5	Investment Strategy Model 1	132
A.2.6	Investment Strategy Model 3	135
A.2.7	Output From Maxflo	137
A.3	Network 3	138
A.3.1	Formula Input File	138
A.3.2	Paths	139
A.3.3	Lower Bound For Original Network	140
A.3.4	Upper Bound For Original Network	145
A.3.5	Investment Strategy Model 1	148
A.3.6	Investment Strategy Model 3	151
A.3.7	Output From Maxflo	154
A.4	Network 4	155
A.4.1	Formula Input File	155

Page

A.4.2 Paths 157

A.4.3 Lower Bound For Original Network 158

A.4.4 Upper Bound For Original Network 161

A.4.5 Investment Strategy Model 1 164

A.4.6 Investment Strategy Model 3 167

A.4.7 Output From Maxflo 171

A.5 Calculating Lower Bound Using Netflo 172

A.5.1 Network 2 172

A.5.2 Network 3 174

A.5.3 Network 4 176

A.6 Calculating Upper Bound Using Microsolv 178

A.6.1 Network 2 178

A.6.2 Network 3 180

A.6.3 Network 4 181

A.7 Investment Strategy Model 1 Using Netflo 183

A.7.1 Network 2 183

A.7.2 Network 3 185

A.7.3 Network 4 187

A.8 Investment Strategy Model 4 Using Netflo 190

A.8.1 Network 2 190

A.8.2 Network 3 192

A.8.3 Network 4 194

A.9 Network A 197

A.9.1 Formula Input File 197

A.9.2 Paths 201

A.9.3 Lower Bound For Original Network 205

A.9.4 Upper Bound For Original Network 211

Page

A.9.5 Investment Strategy Model 1 217

A.9.6 Investment Strategy Model 3 224

A.9.7 Output From Maxflo 242

Appendix B. Formula Version 2.0 User's Manual 243

B.1 Required Equipment 243

B.2 Running the FORMULA 243

B.3 Input 244

B.4 Example 246

B.5 Output 249

B.6 LP/MIP-83 Commands 250

B.7 GINO Commands 250

B.8 Helpful Comments 251

Appendix C. Formula Version 2.0 Source Code 252

Appendix D. Application of Models in a Space Environment 276

D.1 The Space Environment 278

D.1.1 Gravitational Fields. 278

D.1.2 Vacuum. 278

D.1.3 Thermal Radiation. 278

D.1.4 Magnetic Fields. 278

D.1.5 Meteorites and Particles. 279

D.2 Effects of the Space Environment on Satellites 279

D.3 Launching and Transfer Orbits 279

D.4 Improving Reliability in Satellites 280

D.4.1 Preventing Infant Mortality Failures. 280

D.4.2 Providing Reliability Through Quality Control. . . 280

D.4.3 Providing Reliability Through Testing. 281

Page

D.4.4 Testing of One-shot Devices. 281

D.4.5 Use of Redundancy. 281

D.5 Improving Reliability in Satellite Communication Links . . 281

D.6 Improving Reliability in Ground Stations. 282

D.7 Converting the Physical Network to a Graph 282

D.8 Running the Experiment 284

D.9 Analyzing the Results from the Experiment 286

D.10 Deciding on an Improvement Strategy 286

D.11 Summary 287

Appendix E. Branch and Bound Solution to Network 4 288

Bibliography 296

Vita 298

List of Figures

Figure		Page
1.	Stochastic Communication Network Diagram	5
2.	State Diagram	10
3.	Converted Network Using Only Arcs	17
4.	Converted Network with Single Source s and Sink t	18
5.	Network 1	46
6.	Network 2	47
7.	Network 3	48
8.	Network 4	49
9.	Network A	51
10.	Diagram of Experiment	60
11.	Plot of Network 1	92
12.	Plot of Network 2	93
13.	Plot of Network 3	94
14.	Plot of Network 4	95
15.	Plot of Network A	96
16.	Comparison of Network Plots	97
17.	Sample Network	247
18.	Revised Network	248
19.	Typical Space Based Communications Network	277
20.	Graph of a Space Based Network	283
21.	Branch and Bound Tree for Network 4	288

List of Tables

Table		Page
1.	Description of Arcs in Network A	52
2.	Description of Nodes in Network A	53
3.	Dependent Arcs in Network A	54
4.	Dependent Nodes in Network A	54
5.	Original and Final Parameters of Network 1	71
6.	Reliability Improvements For Arcs In Network 1	72
7.	Capacity Improvements For Arcs In Network 1	73
8.	Original and Final Parameters of Network 2	74
9.	Reliability Improvements For Arcs In Network 2	75
10.	Capacity Improvements For Arcs In Network 2	75
11.	Original and Final Parameters of Network 3	77
12.	Reliability Improvements For Arcs In Network 3	77
13.	Capacity Improvements For Arcs In Network 3	78
14.	Original and Final Parameters of Network 4	80
15.	Reliability Improvements For Arcs In Network 4	81
16.	Reliability Improvements For Nodes In Network 4	82
17.	Capacity Improvements For Arcs In Network 4	82
18.	Capacity Improvements For Nodes In Network 4	83
19.	Original and Final Parameters of Network A	84
20.	Reliability Improvements For Arcs In Network A	86
21.	Reliability Improvements For Nodes In Network A	87
22.	Capacity Improvements For Arcs In Network A	87
23.	Lower Bound Comparison	101
24.	Upper Bound Comparison	103

Table		Page
25.	Investment Strategy Model 1 Comparison	104
26.	Optimum Upper Bound Using *Netflow*	106
27.	Investment Strategy Model 3 Versus Investment Strategy Model 4	107
28.	Reliability Improvements For Arcs In Network 2	108
29.	Reliability Improvements For Arcs In Network 3	109
30.	Reliability Improvements For Arcs In Network 4	110
31.	Reliability Improvements For Nodes In Network 4	110
32.	Satellite Nodes	284
33.	Ground Station Nodes	285
34.	Variables for Node 1	290
35.	Variables for Node 2	291
36.	Variables for Node 3	292
37.	Variables for Node 4	293
38.	Variables for Node 5	294
39.	Integer Improvements for Arcs In Network 4	294
40.	Integer Improvements For Nodes In Network 4	295

AFIT/GSO/ENS/90D-10

Abstract

This research investigated the behavior of increasing the reliability of components in stochastic communication networks. A stochastic communication network is one in which the components, links and nodes, are not 100 percent reliable. In addition to the reliability value, each component also has a capacity value that limits the flow through the component. The throughput in a stochastic communication network can be improved by increasing the capacity or the reliability of the individual components.

A set of linear and nonlinear models was developed to determine the best investment strategy for increasing the reliability of individual components. Once the models were developed, an experiment was designed to compare the output of the models to another set of existing models that calculated the best investment strategy for increasing the capacity of individual components. The experiment employed a Prolog program to: enumerate the paths, compute the reliability of each path, and formulate the linear and nonlinear models for direct input into mathematical programming packages. In addition, the linear models were reformulated as networks with side constraints and solved using a network flow package. The models and methodology are general and could easily be modified to fit specific applications.

Improving Reliability In A Stochastic Communication Network

I. Introduction

One concern of managers of stochastic communication networks is the calculation of network throughput under adverse conditions. This calculation is rather straightforward when a network is 100 percent reliable, but becomes almost impossible if some portion of the network can fail without prior warning. This chapter provides a general background of stochastic communication networks and the complications of computing throughput. In addition, the primary research objectives are defined. Assumptions are also listed.

1.1 Background

A stochastic communication network can be diagrammed as a collection of nodes interconnected to each other by a series of arcs, where both the nodes and arcs are randomly susceptible to failure. Each node represents a component in the network, either a transmitter/receiver site or a repeater, and is geographically separated from every other node in the network. The arcs represent communication channels between the nodes. These channels can be telephone lines, microwave links, or satellite links. Because of their fragile electronic components, both nodes and arcs are likely to fail under adverse conditions. Things such as unfavorable weather, natural disasters, terrorist acts, or armed conflict can cause a full or partial failure of a stochastic communication network. It is this reliability factor that concerns managers (23:1).

The limiting throughput, called maximum flow, of a completely reliable network is a function of the capacities of each arc and node. Calculation of the maximum

flow through this type of network has been documented in numerous texts (7). However, as pointed out above, stochastic communication networks are not completely reliable and, therefore, throughput cannot be calculated using conventional methods. Another complication of computing the maximum flow is the enumeration of all the possible states for a network. If a stochastic communication network has n components (arcs and nodes), each with a probability of being either up or down, then there are 2^n possible states. Therefore, the number of states grows exponentially as the number of components increases. This complication makes it nearly impossible to calculate the maximum flow of large networks. The processing time on a computer to enumerate all states would be staggering (23:2).

The Department of Defense (DoD) is concerned with network reliability and survivability, and their impact on information transmission during times of national emergency. Some research has already been accomplished in analyzing the performance of stochastic communication networks. A research team consisting of faculty members in the Operational Sciences department at the Air Force Institute of Technology was formed to assist DoD in analyzing this problem (7). To date, two theses have been completed. The first, written by Yim, used analytical models to calculate the lower and upper bounds of expected maximum flow, and the optimum investment strategy for increasing arc capacity (23:2). The other, prepared by Bailey, concentrated on Monte Carlo simulation to find the expected throughput and expected reliability for a stochastic communication network (4:1).

1.2 Research Objectives

There are two objectives of this research.

1. The first objective is to develop a set of models that provide the optimum investment strategy for improving the reliability of arcs and nodes in a stochastic communications network.

2. The second objective is to take the results from the models formulated in Objective #1 and compare them to a model that calculates the optimum investment strategy for increasing the capacity of arcs and nodes in a stochastic communications network.

1.3 Assumptions

The following assumptions will be made:

1. Component failures are independent.
2. Components either operate or fail.
3. Flow is restricted to one direction.
4. Rerouting of flow is not allowed.
5. Only one commodity flows through the network.

The assumptions are made to simplify the models, but in some cases may not represent a "real world" network. In many networks the components are dependent upon one another, or they are dependent upon a common resource. For example, an electric power station may supply electricity to two or more components in a network. If the power station fails then all the components it supplies power to will also fail. While it is true some components either operate or fail, many will go through various states of degraded performance before failing entirely. The assumptions that flow is in one direction and rerouting is not allowed decreases the flexibility of the network. Modern networks transmit two or more commodities simultaneously, such as voice and data. As seen, the assumptions limit the types of networks that can be analyzed, but the models can be enhanced to handle additional types of networks.

II. Literature Review

This chapter contains a review of the literature applicable to stochastic communication networks. The following sections will discuss notation, network representation, flow models, and network reliability.

2.1 Notation

The following notation will be used in this chapter:

a_{ij} = An element of arc-path incidence matrix: 1 if arc i lies on path j; 0 otherwise

C_j = Choke point capacity of path j, which is equal to $\min[u_i]$ where i is the arcs on path j

f_j = Flow on path j

F_j = Maximum flow on path j

n = Number of arcs in network

p_i = Reliability of component i

P_k = Probability that the failure state is k

q = Number of paths in the network

r = Any designated node; source, intermediate, or sink node

R_j = Reliability of path j

s = Source node

s_k = Failure state k; describes the capacities of the components when the network is in state k

t = Sink node

u_i = Capacity of component i

V_k = Flow through the network from s to t when the failure state is k

2.2 *Network Representation*

The interconnection of links and nodes in a network can be described using either graph theory or matrices. This section describes both methods. In addition, component reliability and capacity are discussed.

2.2.1 Graph Theory. A stochastic communication network can be modeled as a probabilistic graph $G = (V, E)$, where V is a set of nodes, and E is a set of undirected edges. The undirected edges, called links, imply that flow is in both directions. If the flow is restricted to one direction, then E is said to be a collection of directed edges, called arcs (5:521). Each component, either an edge or a node, in the network has two parameters that describes its reliability, p_i, and capacity, u_i. A graphic representation of a stochastic communication network is shown in Figure 1. The notation used in Figure 1 is p_i, u_i, where $u_i = *$ implies that the link/node has infinite capacity.

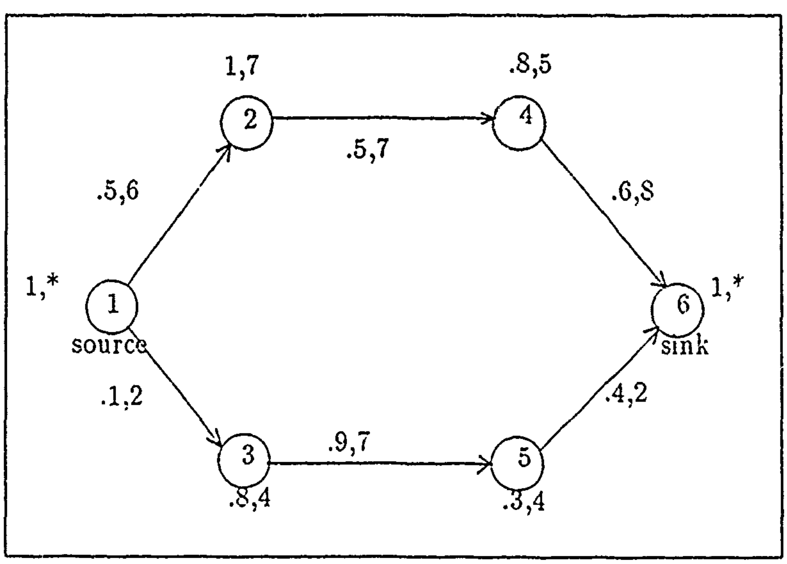


Figure 1. Stochastic Communication Network Diagram

2.2.2 Matrices. Another method of representing a stochastic communication network is the use of either an node-arc incidence matrix or an arc-path incidence matrix. In the node-arc incidence matrix, each row corresponds to an arc and every column represents a node. Each entry in the matrix, denoted by e_{ik}, is either a 1, -1, or 0. A 1 implies that arc i ends at node k, while a -1 indicates i starts at k, and finally 0 is used to show no connection exists. In contrast, the arc-path incident matrix uses rows to represent arcs, but the columns correspond to paths in the network. The arc-path incident matrix uses a_{ij} to indicate whether an arc lies on a particular path. A 1 implies that the arc i lies on path j, and 0 indicates otherwise (23:7).

2.2.3 Component Reliability and Capacity. The reliability of a component can be described in many different ways. Brecht and Colbourn (5:521) use p_i to describe the probability that component i operates to some level of desired operation. El-Sheikh and Cheston (10) state that p_i can be approximated by the following relation:

$$p_i = \frac{mut_i}{mdt_i + mut_i} \tag{1}$$

where

mut_i = the mean amount of time that component i is up

mdt_i = the mean amount of time that component i is down.

The probabilities are limiting state probabilities, which hold as time approaches infinity (10:279).

Yim (23:1) uses p_i to denote the probability that component i survives a catastrophic event, such as inclement weather or armed conflict. Both Brecht/Colbourn and Yim assume that components fail independently of one another. In addition, both sources assume that p_i represents the probability that the event either happens, or it does not.

In addition to reliability, each component has a capacity, u_i, associated with it. This parameter describes the maximum amount of flow that component i can handle. Capacity is normally measured in units/time, such as cars per hour or the number of simultaneous telephone conversations per minute. The law of conservation of flow states that the amount of flow leaving a node must be equal to the sum of the flows entering the node (23:7). This can easily be described using Kirchhoff's conservation laws. Let s designate the source node and t the sink node of a network. Now let the vector $\phi = (\phi_1, \phi_2, \ldots, \phi_M)$ where:

M	=	Number of edges
ϕ_u	=	The amount of resources used on edge u ($u = 1, 2, \ldots, M$)
$w^+(r)$	=	edges leaving node r
$w^-(r)$	=	edges entering node r
$w^+(s)$	=	edges leaving source node s
$w^-(t)$	=	edges entering sink node t

Resources are defined to be transmission or communication facilities. By limiting flow to one direction through each edge, then Kirchhoff's law holds at each node in the network, i.e.,

$$0 = \sum_{u \in \omega^+(r)} \phi_u - \sum_{u \in \omega^-(r)} \phi_u \tag{2}$$

for any node , r, except s and t, and

$$v(\phi) = \sum_{u \in \omega^+(s)} \phi_u = \sum_{u \in \omega^-(t)} \phi_u \tag{3}$$

for s and t. Note, $v(\phi)$ is the total amount of flow leaving s and entering t, and is called the value of flow ϕ. Equations 2 and 3 are Kirchhoff's laws, and describe the basic relations defining a flow with a source and sink (17:315).

2.3 *Flow Models*

Minoux defines four basic types of flow models (17:315-316). Each is described below.

1. Single-Commodity Flows. In single-commodity flows there is only one flow requirement between s (source) and t (sink). This can be written as

   $$A.\phi = v(\phi).b$$

   where A is the $(N \times M)$ node-arc incident matrix of graph $G = (V, E)$, and b is an N-vector with all indexes equal to 0 except $b_s = 1$ and b_t = -1. Figure 1 is an example of a single commodity flow network.

2. Multicommodity Flows. Multicommodity flows consist of many pairs of nodes which communicate simultaneously. Take for example a network that has K individual single-commodity requirements, where $k = (1, 2, \ldots, K)$. Each $s(k)$ and $t(k)$ denote the source and sink nodes of a single-commodity flow network All single-commodity flow networks ($s(k)$ to $t(k)$) are operating simultaneously.
3. Nonsimultaneous Flows. The nonsimultaneous flow network has the same requirement as the multicommodity flow network described above (k simultaneous single-commodity flows). However, in a nonsimultaneous flow network, the single-commodity flows are independent of each other.
4. Multiterminal Flows. The multiterminal (single-commodity) flow network is a special case of the single-commodity flow network, where there may exist a single source s and multiple sinks t. or multiple sources s and a single sink t.

Algebraic representations exist for the multicommodity, nonsimultaneous (single-commodity), and multiterminal flow networks, but they are not necessary for this research (see assumptions in Chapter I).

2.4 *Network Reliability*

The existing research in network reliability has been divided between two areas. The first area is networks with binary components, where the components either operate or fail. The second is networks with multimode components. Multimode components have various levels of failure. Another area that is starting to get attention is networks with dependent components.

2.4.1 *Binary Components.*

The reliability of a stochastic communication network is a function of the reliability of its individual components. Sancho (19) uses dynamic programming to calculate the most reliable path. Let

R_i = the reliability of path of maximum reliability from source, s, to sink, t

p_{ij} = the reliability of the arc going from node i to node j

Sancho then computes R_i using the principle of optimality

$$R_i = max[p_{ij}R_j] \tag{4}$$

for $i \neq j$, and $R_t = 1$ as the boundary condition. The solution to this problem yields the path with the highest reliability. Once R, path with highest probability, has been found, then the expected capacity, F_i, through path i is computed by

$$F_i = R_iC_i \tag{5}$$

where

$$C_i = \text{the choke point capacity on path } i$$

and the maximum expected capacity path is determined by calculating the path with the largest F_i (19:261-262).

Kubat (12) computes the probability of each state, P_k, where k is a specific state. He then uses P_k to compute the expected maximum flow for a system that

has M possible states:

$$Expected\ \ Maximum\ \ Flow = \sum_{k=0}^{M} V_k \cdot P_k \tag{6}$$

where V_k is the flow through the system in state k (12:308-309). This method requires the calculation of P_k for each state, k, the system can be in. If the system has n components, then the number of possible states is equal to 2^n. For large networks this calculation is almost impossible (23:2). A medium size network of 50 components, both links and nodes, would result in approximately $1.12X10^{15}$ possible states! This phenomenon is called the "State-Space Explosion" problem (14:493). Figure 2 contains a sample network, and all the possible states it can be in. Assume that only the arcs are subject to failure in the sample network. Since there are 3 arcs, the total number of states is 2^3, which equals 8.

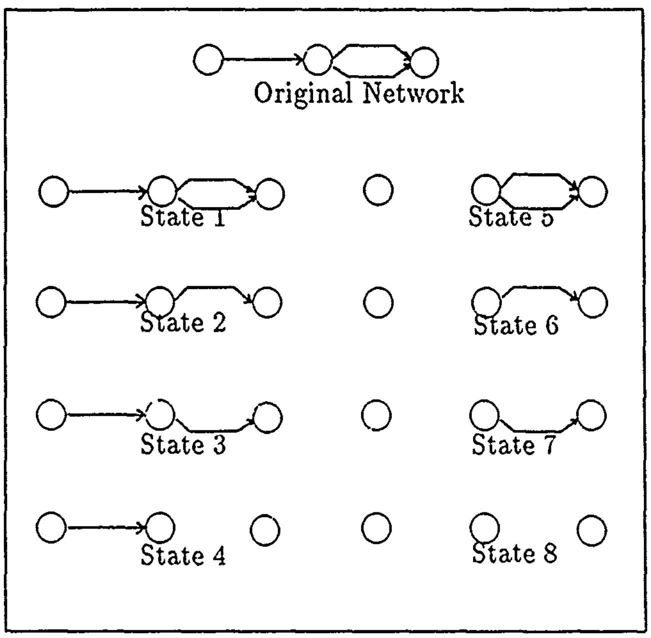


Figure 2. State Diagram (23:14)

Li and Silvester (14) developed an algorithm called *Order* to calculate the most probable states of a network consisting of unreliable components. They recognized the "State-Space Explosion" problem, and attempted to solve it by restricting their work to the most probable states. By doing so, they were able to compute the lower and upper bounds of network reliability, and get a good approximation of system performance. They assumed n components either operated or failed, with probability of p_i and q_i respectively, where q_i = 1 - p_i. Links were then ordered $R_1 \geq R_2 \geq \cdots \geq R_n$ where $R_i = q_i/p_i$. It should be noted that $\frac{1}{2} \leq p_i \leq 1$, for i = 1, 2, ... , n. The probability of state s_k is computed by

$$P(s_k) = \prod_i p_i(\frac{q_i}{p_i})^{T_i(s_k)} \tag{7}$$

for

$$T_i(s_k) = \begin{cases} 0 & \text{if link } i \text{ operates in state } s_k \\ 1 & \text{otherwise} \end{cases}$$

The probability of the states are then ordered P(s_1) $\geq$ P(s_2) $\geq \cdots \geq$ P(s_n). The most reliable state is s_1 which corresponds to no failures, where as state s_n represents total failure (14:494). It is now possible to study the most reliable states, and thereby eliminate the "State-Space Explosion" problem described earlier. The *Order* algorithm is described in detail by Li and Silvester (14).

Yim, Chan, and Marsh (24) were also concerned with computing the lower and upper bounds of network reliability. They assumed components were independent and defined the reliability of a path j to be

$$R_j = \prod_{i \in A_j} p_i \tag{8}$$

where A_j is a path in the network, and p_i is the reliability of each arc on the path. The probability of a stochastic communication network is computed for state s_k by

$$P(s_k) = (\prod_{i \in A} p_i)(\prod_{i \in A\prime} [1 - p_i]) \tag{9}$$

where A is the set of operating arcs and $A\prime$ is the set of failed arcs. The lower bound of reliability , R_L, can be computed using P(s_k), as seen below.

$$R_L = \sum_{k=1}^{r} P(s_k) R(s_k) \tag{10}$$

where r is the set of most probable states, and $R(s_k)$ is the reliability of the network in state s_k. The upper bound of reliability is

$$R_U = R_L + [1 - \sum_{k=1}^{r} P(s_k)] \tag{11}$$

It is important to note that $r \ll 2^n$. As can be seen, the use of lower and upper bounds eliminates the "State-Space Explosion" problem. Yim, Chan, and Marsh go on to define two important measures of stochastic communication networks: reliability and vulnerability. Reliability of a network is defined to be

$$R = 1 - \frac{\sum_{j=1}^{r^*} R_j}{\sum_{j=1}^{r\prime} R_j} \tag{12}$$

and vulnerability is

$$T = 1 - \frac{\sum_{j=1}^{r^*} R_j u_j}{\sum_{j=1}^{r\prime} R_j u_j} \tag{13}$$

where r^* is the set of paths that are actually used, and $r\prime$ is a set of all paths from source, s, to sink, t (24:5-12).

Aneja and Nair (2) computed the reliability of a multicommodity flow network. The details are omitted here because single-commodity flow networks are assumed in Chapter I. Aneja and Nair's works are mentioned here for the benefit of the reader.

2.4.2 Multimode Components. Components with multiple states of failure have also been studied by researchers. Their work has concentrated on multimode components, which operate in one of $N(N \geq 3)$ modes. They have recognized the fact that a component undergoes various levels of degraded service before it fails completely.

Chiou and Li (8) extended the work by Li and Silvester (14) to include networks with multimode components. The results of Chiou and Li's work was a modified version of the *Order* algorithm, called *Order-M.* Chiou and Li dropped the assumption that components either operated or failed. Each component is associated with a known reliability. Let there be M links in the network, and N (0, 1, 2, ... , N-1) modes of operation for each link. Then the jth mode of arc i can be denoted by C_i^j, and the probability of arc i being in mode j is

$$p_i^j = Pr(C_i^j) \tag{14}$$

If link failures are independent then

$$\sum_{j=0}^{N-1} p_i^j = 1 \tag{15}$$

for $i = 1, 2, \ldots, M$. The modes are then renamed such that the mode probabilities are in decreasing order, i.e., $p_i^0 \geq p_i^1 \geq \cdots \geq p_i^{N-1}$ for all i. Now the reliability of each state can be computed as

$$\prod_{i}^{M} p_i^j \tag{16}$$

for $j = 0, 1, \ldots, N-1$. The reliability of all the states is then ordered from highest to lowest, where highest indicates the most reliable state. It is now possible to study the behavior of the network by considering only the most reliable states, as they will have the most influence on expected throughput (8:1156-1157). The task of identifying all the states is not simple, as was indicated earlier. An in-depth understanding of

Order-M is not necessary, because the assumptions in Chapter I assume components either operate or fail.

Yang and Kubat (22) developed another algorithm to calculate the most probable states when a network has multimode components. Their algorithm, which was unnamed, is superior in speed and flexibility to *Order-M*. Yang and Kubat's algorithm generates a minimal set of probable states, called Ω_ϵ. The subscript ϵ comes from the minimal coverage requirement (1 - ϵ) where $\sum_{k=1}^{m} P(s_k) \geq (1 - \epsilon)$ (notation is same as Li and Silvester). The minimal coverage requirement is established by the researcher, and is used to determine the number of states to be computed (22:535).

2.4.3 Dependent Failures. Up to this point all literature assumed reliability was independent from one component to another. However, for most real world problems this assumption does not hold. Lam and Li (13) have developed a model called the Event-Based Reliability Model (EBRM) to calculate the reliability of communications networks that have links which are statistically dependent. The EBRM uses conditional probability to compute the networks reliability. The details of how EBRM works will not be discussed, because as stated in Chapter I, all links and nodes are assumed to be independent in this research.

2.5 Lower and Upper Bounds of Expected Maximum Flow

The calculation of expected maximum flow (Equation 6) for a large network is mathematically infeasible because of the "State-Space Explosion" problem. As a result, many researchers have elected to calculate the lower and upper bounds of the expected maximum flow. While the lower and upper bounds will not give the exact solution, they will provide a worst and best case respectively.

2.5.1 Lower Bound of Expected Maximum Flow. The lower bound of expected maximum flow provides a pessimistic appraisal of the throughput in a stochastic communication network (23:16). Therefore, the lower bound is a worst case sce-

nario, and is used when the rerouting of flow is not allowed (6:441). Aneja and Nair (2) refer to the lower bound as maximum expected flow. Aneja and Nair (2) developed a linear program for computing lower bound when rerouting was prohibited. Yim (23) used Aneja and Nair's formulation exactly. Carey and Hendrickson used a much simpler formulation, the *minimum-cost flow routing problem*, but their results were not as good as that of Aneja and Nair (6:439-440). A formulation using Yim's notation is given in the following section.

2.5.1.1 Formulation. Let R_j equal the reliability of path j (j = 1, 2, ..., q). As indicated by Equation 8, the reliability of path j can be computed by

$$R_j = \prod_{i \in A_j} p_i$$

where A_j is a path in the network. Now let f_j be the flow on path j. The sum of expected flows on all paths from source, s, to sink, t, is given by

$$\sum_{j=1}^{q} R_j f_j \tag{17}$$

Now let a_{ij} equal 1 if arc i (i = 1, 2, ..., n) lies on path j, and 0 otherwise. If rerouting is not allowed then the lower bound can be formulated as follows:

$$\text{Max } \sum_{j=1}^{q} R_j f_j$$

s.t.

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i \text{ for } i = 1, 2, \ldots, n$$

$$f_j \geq 0$$

Note, u_i is the capacity of arc i (23:16-18).

2.5.2 Upper Bound of Expected Maximum Flow. The upper bound of expected maximum flow provides an optimistic estimate of the throughput. Yim states that "knowledge of the upper bound is not as critical as that of the lower bound,

but that the upper bound is useful in some applications" (23:18). Yim's formulation of upper bound is given below.

2.5.2.1 Formulation. The upper bound formulation assumes the most optimistic scenario, that all arcs are fully operational (24:6), i.e., all paths are up. Let f_j be the flow on path j (j = 1, 2, ..., q). Therefore the formulation of the upper bound is

$$\text{Max } \textstyle\sum_{j=1}^{q} f_j$$

s.t.

$$\textstyle\sum_{j=1}^{q} a_{ij} f_j \leq e(u_i) \quad \text{for } i = (1, 2, \ldots, n)$$

$$f_j \geq 0$$

where $e(u_i) = p_i \cdot u_i$ (23:37).

2.6 Formula

Formula is a Prolog-based computer program written by Yim (23) to generate a set of linear models that can then be solved using a commercial linear programming package (23:28). The program runs on any *IBM*-compatible personal computer, and requires an Arity/Prolog Version 5.0 compiler. To use *Formula*, the user must first create a file that contains a description of the stochastic communication network to be analyzed. *Formula* then reads the file and generates a linear program, complete with objective function and constraints for input into the LP/MIP-83 mathematical programming package. Lastly, the linear program is solved by LP/MIP-83. The solution generated by LP/MIP-83 depends on the the linear program that was formulated. *Formula* will formulate the models for calculating the lower and upper bounds of flow, the maximum flow, and the optimum investment strategy for increasing arc capacity. In addition to the linear programs, *Formula* will create an output file which contains a list of all the possible paths and their respective reliabilities (23:48-54).

Formula makes use of the fact that any node can be replaced by two nodes joined together by one dummy arc. The dummy arc will have the same reliability and capacity as the node being replaced. By performing this simple step, it is now possible to describe the network using only arcs. Figure 3 shows the network from Figure 1 after its conversion. The nodes in Figure 3 are not stochastic (they are deterministic) since they have a reliability of 1 and an infinite capacity.

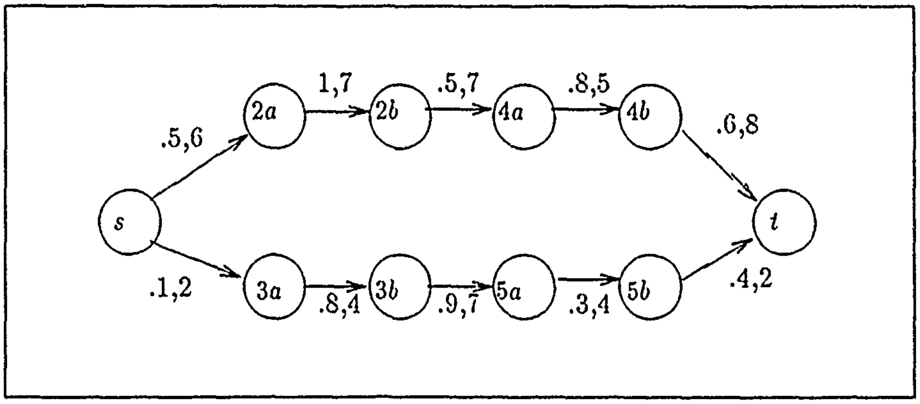


Figure 3. Converted Network Using Only Arcs

In addition, if there is more than one source or sink node in the network, and it is possible for flow to go from any source to any sink, i.e., single-commodity, then the network can be converted to a network containing a single source and a single sink. This is accomplished by adding a new source node, s, or sink node, t, and adding arcs from the newly created node to the existing source/sink nodes. The new arcs will have a reliability of 1 and unlimited capacity (23:7-12). Figure 4 shows a simple example of converting a network that had two source and two sink nodes into a network with a single source and sink.

Before using *Formula*, the user must replace all stochastic nodes with dummy arcs, and ensure the network has a single source and single sink. This converted network is then input into the file that describes the stochastic communication network. At the user's request, *Formula* will read the file and provide a menu allowing

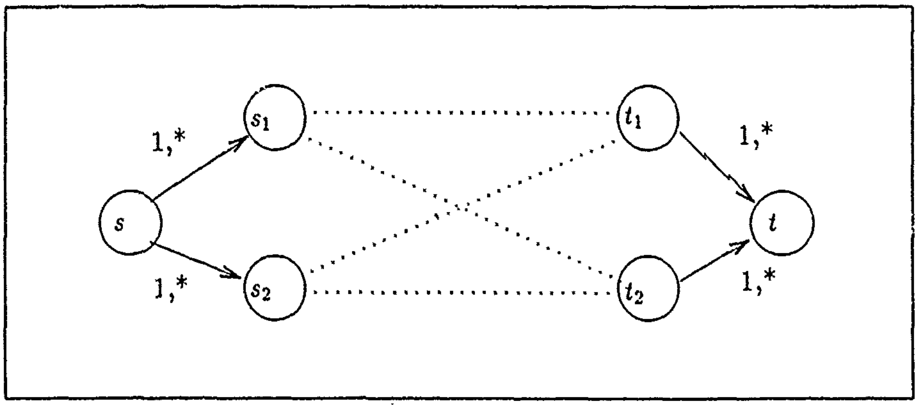


Figure 4. Converted Network with Single Source s and Sink t

the user to enter his request. The user can request that *Formula* generate a list of all paths and their reliabilities, formulate the models to compute lower/upper bounds and maximum flow, or formulate the models to generate the optimum investment strategy for increasing arc capacity. The output from *Formula* is written to files which can then be input into LP/MIP-83 (23:99-127). A users manual for *Formula* is contained in Appendix B.

2.7 Simulation

Bailey (4) used Monte Carlo simulation to calculate the expected maximum flow in a stochastic communication network. His simulation, called *Maxflo*, made use of variance and bias reduction techniques of control variates and antithetic random numbers, and introduced the technique of Response Surface Methodology (RSM) to the output analysis (4:1). In addition, *Maxflo* will calculate the expected reliability of the network. The simulation program allowed the user to input a description of the stochastic communication network, and then used a random number generator to determine which arcs would fail, and which would operate. The expected maximum flow would then be computed on the network. RSM was used to provide an algebraic description of network flow and reliability, and how individual components influence

expected maximum flow (4:5.1-5.2). For a complete understanding of RSM, and its application to stochastic communication networks, the reader should consult Bailey's thesis.

2.8 Summary

The majority of the literature on stochastic communication networks is devoted to solving the "State-Space Explosion" problem. Because the number of states in a network is on the order of 2^n, it is almost impossible to calculate the expected maximum flow through the network. There are various algorithms and methods in use to estimate the maximum expected flow, and each is dependent on the definition of network reliability. Most sources computed network reliability using the following assumptions:

- Component failures are independent.
- Components are binary; they either operate or fail.
- Network flow is single-commodity.
- Flow is restricted to one direction.
- The network does not contain any cycles.
- Rerouting of flow is not allowed.

Probabilities were used by all sources to describe component, path, and network reliability. A small amount of research has been done using networks that have multimode or dependent components.

The Prolog-based *Formula* is excellent for enumerating the paths in a stochastic communication network. This program will compute the reliability of each path, and formulate the linear program for identifying the components which are best candidates for improving capacity. The methodology outlined in Chapter IV will describe the use of *Formula* in this thesis.

Monte Carlo simulation has been used to analyze the performance of stochastic communication networks. The *Maxflo* simulation written by Bailey implemented Monte Carlo simulation to calculate the expected maximum flow, and employed response surface methodology to create an algebraic description of network flow and reliability. Chapter IV outlines how *Maxflo* will be used in this research.

III. *Formulations of Models*

This chapter describes the models used in determining the best investment strategies for increasing capacity and reliability in a stochastic communication network. The models will determine which arcs should have their capacity or reliability increased. The models used to determine the best investment strategy for improving reliability will also evaluate potential arcs which may be added to the network. Once an arc has been added, it is a candidate for subsequent increases in reliability. The formulation of the models in this chapter accomplishes research objective #1.

There are seven models presented in this chapter. Two of the models were originally developed by Capt Eugene Yim for his thesis, and their objective functions maximize the lower bound by increasing capacity on existing arcs. The first model, called Investment Strategy Model 1, calculates an optimal lower bound using continuous variables, while the other model, Investment Strategy Model 2, allows for only integer solutions (23:38-41). Two new models are presented to determine the optimum investment strategy for increasing reliability using continuous variables. They are called Investment Strategy Model 3 and Investment Strategy Model 4. Investment Strategy Model 3 will calculate the optimum investment strategy for increasing reliability by maximizing the lower bound of expected maximum flow, whereas Investment Strategy Model 4 maximizes the upper bound by increasing reliability. Investment Strategy Models 5 and 6 use integer variables. Investment Strategy Model 5 optimizes the lower bound by increasing arc reliability, and Investment Strategy Model 6 optimizes the upper bound by increasing arc reliability. Investment Strategy Model 7 optimizes the lower bound by using continuous variables to increase capacity and reliability simultaneously.

3.1 *Assumptions and Restrictions*

Chapter I contains a set of assumptions made on the networks to be analyzed. In addition, the following assumptions and restrictions are placed on the models:

1. Reliability will be increased in increments of 0.1. This value was picked for convenience, but any positive value less than 1 can be used depending on the application.
2. The optimum locations for potential arcs are known. The model does not pick the best location for potential arcs. The user of the model must determine where potential arcs are to be located.
3. The initial reliability of potential arcs is set to 0. Setting the reliability to 0 implies the new arc does not yet exist and prohibits flow through the arc.
4. The lower and upper limits of reliability for an arc is 0 and 1, respectively.
5. Any node that has a reliability less than 1 or a finite capacity must be converted into an arc connected by two dummy nodes (see Section 2.6).

3.2 *Decision Variables*

The following decision variables will be used to formulate the models:

R_j = the reliability of path j

f_j = the flow through path j

X_i = the number of .1 increments of reliability added to arc i

P_i = the initial reliability of arc i

d_i = the increase in capacity of arc i

u_i = the initial capacity of arc i

a_{ij} = equals 1 if arc i lies on path j; 0 otherwise

α_{ci} = the cost of increasing capacity by 1 for arc i

α_{ri} = the cost of increasing reliability by .1 for arc i

b_i = the predetermined amount of capacity increase, per investment, in arc i

g_i = the number of investments in capacity for arc i

β = the total amount of the budget

n = the total number of arcs in the network

q = the total number of paths in the network

The variables $P_i, u_i, \alpha_{ci}, \alpha_{ri}, \beta, n$, and q are constants. The budget amount, β, plus the α_{ci} and α_{ri} costs are the same for all seven models.

3.3 Formulation of Investment Strategy Model 1

This section contains the formulation for Investment Strategy Model 1. The model will maximize the lower bound of expected maximum flow by increasing the capacity of arcs in the network. The variables denoting the amount of capacity increase are continuous. There is no limit to the amount of capacity that can be added to an existing arc.

3.3.1 Objective Function. The lower bound is the sum of expected flows for each path in the network. The expected flow through path j ($j = 1, 2, \ldots, q$) is $R_j f_j$, where R_j is a product of arc reliabilities

$$R_j = \prod_i P_i \quad \text{for all arcs } i \ (i = 1, 2, \ldots, n) \text{ on path } j$$

and f_j is the flow through path j. The objective is to maximize the lower bound by increasing the capacity of each path, therefore the objective function is

$$\text{Max} \sum_{j=1}^{q} R_j f_j$$

The reader should notice that R_j is a constant in Investment Strategy Model 1.

3.3.2 Constraints. There are two types of constraints in the model. They are:

1. Flow capacity constraints for each path j.
2. A budget constraint indicating the amount of funds available for investment, and the cost of increasing capacity on each arc i by 1 unit.

Let a_{ij} equal 1 if arc i lies on path j, and 0 otherwise. Using this definition, the constraints for flow capacity are

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i + d_i$$

Where u_i is the capacity of arc i, and d_i is the number of units of capacity added to arc i.

The budget constraint is

$$\sum_{i=1}^{n} \alpha_{ci} d_i \leq \beta$$

Where α_{ci} is the cost of increasing the capacity of arc i by 1, and β is the total amount of the budget available to be invested.

3.3.3 Model. The model for Investment Strategy Model 1 is

$$\text{Max } \sum_{j=1}^{q} R_j f_j \qquad \text{where } R_j = \prod_i P_i$$

s.t.

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i + d_i$$

$$\sum_{i=1}^{n} \alpha_{ci} d_i \leq \beta$$

$$R_j, u_i, d_i, f_j, \alpha_{ci}, \beta \geq 0 \text{ and } a_{ij} = 0, 1$$

An important fact about Investment Strategy Model 1 is that the objective function and the constraints are linear, therefore the model can be solved using a conventional linear programming package (23:38).

3.4 Formulation of Investment Strategy Model 2

Investment Strategy Model 2 will maximize the lower bound by increasing arc capacity, but unlike Investment Strategy Model 1 it requires the solution to consist of only positive integer values.

3.4.1 Objective Function. The lower bound is the sum of expected flows for each path in the network. The expected flow through path j ($j = 1, 2, \ldots, q$) is $R_j f_j$, where R_j is a product of arc reliabilities

$$R_j = \prod_i P_i \quad \text{for all arcs } i \ (i = 1, 2, \ldots, n) \text{ on path } j$$

and f_j is the flow through path j. The objective is to maximize the lower bound by increasing the capacity of each path, therefore the objective function is

$$\text{Max} \sum_{j=1}^{q} R_j f_j$$

The reader should notice that R_j is a constant in Investment Strategy Model 2.

3.4.2 Constraints. There are two types of constraints in the model. They are:

1. Flow capacity constraints for each path j.
2. A budget constraint indicating the amount of funds available for investment, and the cost of increasing capacity on each arc i by 1 unit.

Let a_{ij} equal 1 if arc i lies on path j, and 0 otherwise. Using this definition, the constraints for flow capacity are

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i + b_i g_i$$

Where u_i is the capacity of arc i, b_i is the predetermined amount of capacity increase on arc i, and g_i is the number of investments made in arc i.

The budget constraint is

$$\sum_{i=1}^{n} \alpha_{ci} b_i g_i \leq \beta$$

Where α_{ci} is the cost of increasing the capacity of arc i by 1 unit, and β is the total amount of the budget available to be invested.

3.4.3 Model. The model for Investment Strategy 2 is

$$\text{Max } \sum_{j=1}^{q} R_j f_j \qquad \text{where } R_j = \prod_i P_i$$

s.t.

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i + b_i g_i$$
$$\sum_{i=1}^{n} \alpha_{ci} b_i g_i \leq \beta$$
$$R_j, u_i, b_i, f_j, g_i, \alpha_{ci}, \beta \geq 0,\ a_{ij} = 0, 1, \text{ and } g_i = \text{integer}$$

Investment Strategy Model 2 must be solved using a package the supports mixed integer solutions (23:40).

3.5 Formulation of Investment Strategy Model 3

Investment Strategy Model 3 will use continuous variables to maximize the lower bound of expected maximum flow by increasing the reliability of arcs. The model will determine if potential arcs should be added to the network. Potential arcs have an initial reliability of 0 and a capacity of 5.

3.5.1 Objective Function. The lower bound is the sum of expected flows for each path in the network. The expected flow through path j ($j = 1, 2, \ldots, q$) is $R_j f_j$, where R_j is a product of arc reliabilities

$$R_j \;=\; \prod_i (P_i + .1 \cdot X_i) \quad \text{for all arcs } i \ (i = 1, 2, \ldots, n) \text{ on path } j$$

and f_j is the flow through path j. The objective is maximize the lower bound by increasing the reliability, R_j, of each path. therefore the objective function is

$$\text{Max } \sum_{j=1}^{q} R_j f_j$$

The objective function is nonlinear since R_j is a product of X_i's.

3.5.2 Constraints. There are three types of constraints in the model. They are:

1. Constraints to ensure arc reliabilities are ≤ 1.
2. Flow capacity constraints for each path j.
3. A budget constraint indicating the amount of funds available for investment, and the cost of increasing reliability on each arc i.

The constraints to ensure reliabilities are less than 1 are

$$P_i + .1 \cdot X_i \leq 1$$

There is a reliability constraint for each arc in the network.

Let a_{ij} equal 1 if arc i lies on path j, and 0 otherwise. Using this definition, the constraints for flow capacity are

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i$$

Where u_i is the capacity of arc i.

The budget constraint is

$$\sum_{i=1}^{n} \alpha_{ri} X_i \leq \beta$$

Where α_{ri} is the cost of increasing the reliability of arc i by .1, and β is the total amount of the budget available to be invested.

3.5.3 Model. Combining the objective function and the constraints results in the following nonlinear model:

$$\text{Max } \sum_{j=1}^{q} R_j f_j \quad \text{where } R_j = \prod_i (P_i + .1 \cdot X_i)$$

s.t.

$$P_i + .1 \cdot X_i \leq 1 \quad i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i$$

$$\sum_{i=1}^{n} \alpha_{ri} X_i \leq \beta$$

$$X_i, P_i, f_j, \alpha_{ri}, \beta \geq 0 \text{ and } a_{ij} = 0, 1$$

The model will be referred to as Investment Strategy Model 3, and it's solution requires a nonlinear programming package.

3.6 Formulation of Investment Strategy Model 4

The upper bound of expected maximum flow will be used as the measure of network throughput in Investment Strategy Model 4. The model will use continuous variables to increase arc reliability. In addition, potential arcs will be evaluated to see if they need to be added to the network.

3.6.1 Objective Function. The upper bound is the sum of the flows for each path in the network. It assumes all paths are up. The objective is to maximize the upper bound by increasing the flow on path j ($j = 1, 2, \ldots, q$). Therefore the objective function is

$$\text{Max } \sum_{j=1}^{q} f_j$$

The objective function is linear.

3.6.2 Constraints. There are three types of constraints in the model. They are:

1. Constraints to ensure arc reliabilities are ≤ 1.
2. Flow capacity constraints for each path j.

3. A budget constraint indicating the amount of funds available for investment, and the cost of increasing reliability on each arc i.

The constraints to ensure reliabilities are less than 1 are

$$P_i + .1 \cdot X_i \leq 1$$

There is a reliability constraint for each arc i ($i = 1, 2, \ldots, n$) in the network.

Let a_{ij} equal 1 if arc i lies on path j, and 0 otherwise. Using this definition, the constraints for flow capacity are

$$\sum_{j=1}^{q} a_{ij} f_j \leq (P_i + .1 \cdot X_i) u_i$$

Where the right hand side of the constraint is the expected capacity of the bottleneck arc on path j.

The budget constraint is

$$\sum_{i=1}^{n} \alpha_{ri} X_i \leq \beta$$

Where α_{ri} is the cost of increasing the reliability of arc i by .1, and β is the total amount of the budget available to be invested.

3.6.3 Model. Combining the objective function and the constraints results in the following linear model:

$$\text{Max } \sum_{j=1}^{q} f_j$$

s.t.

$$P_i + .1 \cdot X_i \leq 1 \qquad i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{q} a_{ij} f_j \leq (P_i + .1 \cdot X_i) u_i$$

$$\sum_{i=1}^{n} \alpha_{ri} X_i \leq \beta$$

$$X_i, P_i, f_j, \alpha_{ri}, \beta \geq 0 \text{ and } a_{ij} = 0, 1$$

The model will be referred to as Investment Strategy Model 4, and can be solved using a linear programming package.

3.7 *Formulation of Investment Strategy Model 5*

Investment Strategy Model 5 will maximize the lower bound of expected maximum flow by increasing the reliability of arcs in discrete amounts. The model will also determine if potential arcs should be added to the network.

3.7.1 Objective Function. The lower bound is the sum of expected flows for each path in the network. The expected flow through path j ($j = 1, 2, \ldots, q$) is $R_j f_j$, where R_j is a product of arc reliabilities

$$R_j \quad = \quad \prod_i (P_i + .1 \cdot X_i) \quad \text{for all arcs } i \ (i = 1, 2, \ldots, n) \text{ on path } j$$

and f_j is the flow through path j. The objective is to maximize the lower bound by increasing the reliability, R_j, of each path. Therefore the objective function is

$$\text{Max} \sum_{j=1}^{q} R_j f_j$$

The objective function is nonlinear since R_j is a product of X_i's.

3.7.2 Constraints. There are three types of constraints in the model. They are:

1. Constraints to ensure arc reliabilities are ≤ 1.
2. Flow capacity constraints for each path j.
3. A budget constraint indicating the amount of funds available for investment, and the cost of increasing reliability on each arc i.

The constraints to ensure reliabilities are less than 1 are

$$P_i + .1 \cdot X_i \leq 1$$

There is a reliability constraint for each arc in the network.

Let a_{ij} equal 1 if arc i lies on path j, and 0 otherwise. Using this definition, the constraints for flow capacity are

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i$$

Where u_i is the capacity of arc i.

The budget constraint is

$$\sum_{i=1}^{n} \alpha_{ri} X_i \leq \beta$$

Where α_{ri} is the cost of increasing the reliability of arc i by .1, and β is the total amount of the budget available to be invested.

3.7.3 Model. Combining the objective function and the constraints results in the following nonlinear model:

$$\text{Max } \sum_{j=1}^{q} R_j f_j \quad \text{where } R_j = \prod_i (P_i + .1 \cdot X_i)$$

s.t.

$$P_i + .1 \cdot X_i \leq 1 \quad i = 1, 2, \ldots, n$$
$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i$$
$$\sum_{i=1}^{n} \alpha_{ri} X_i \leq \beta$$
$$P_i, f_j, X_i, \alpha_{ri}, \beta \geq 0,\ a_{ij} = 0, 1, \text{ and } X_i = \text{integer}$$

The model will be referred to as Investment Strategy Model 5, and it's solution requires the use of an integer programming algorithm such as Branch and Bound. The calculation of the solution at each node in the Branch and Bound algorithm will require a nonlinear programming package.

3.8 Formulation of Investment Strategy Model 6

The upper bound will be used as the measure of network throughput in Investment Strategy Model 6. The model will maximize the upper bound by increasing arc reliability in discrete amounts. In addition, potential arcs will be evaluated to see if they need to be added to the network.

3.8.1 Objective Function. The upper bound is the sum of the flows for each path in the network. It assumes all paths are up. The objective is to maximize the upper bound by increasing the reliability of the bottleneck arc along path j ($j = 1, 2, \ldots, q$). Therefore the objective function is

$$\text{Max} \sum_{j=1}^{q} f_j$$

The objective function is linear.

3.8.2 Constraints. There are three types of constraints in the model. They are:

1. Constraints to ensure arc reliabilities are ≤ 1.
2. Flow capacity constraints for each path j.
3. A budget constraint indicating the amount of funds available for investment, and the cost of increasing reliability on each arc i.

The constraints to ensure reliabilities are less than 1 are

$$P_i + .1 \cdot X_i \leq 1$$

There is a reliability constraint for each arc i ($i = 1, 2, \ldots, n$) in the network.

Let a_{ij} equal 1 if arc i lies on path j, and 0 otherwise. Using this definition, the constraints for flow capacity are

$$\sum_{j=1}^{q} a_{ij} f_j \leq (P_i + .1 \cdot X_i) u_i$$

Where the right hand side of the constraint is the expected capacity of the bottleneck arc on path j.

The budget constraint is

$$\sum_{i=1}^{n} \alpha_{ri} X_i \leq \beta$$

Where α_i is the cost of increasing the reliability of arc i by .1, and β is the total amount of the budget available to be invested.

3.8.3 Model. Combining the objective function and the constraints results in the following linear model:

$$\text{Max } \sum_{j=1}^{q} f_j$$

s.t.

$$P_i + .1 \cdot X_i \leq 1 \qquad i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{q} a_{ij} f_j \leq (P_i + .1 \cdot X_i) u_i$$

$$\sum_{i=1}^{n} \alpha_{ri} X_i \leq \beta$$

$$P_i, f_j, X_i, \alpha_{ri}, \beta \geq 0, \ a_{ij} = 0, 1, \text{ and } X_i = \text{integer}$$

The model will be referred to as Investment Strategy Model 6, and must be solved using a mixed integer programming package.

3.9 Formulation of Investment Strategy Model 7

Investment Strategy Model 7 will use continuous variables to maximize the lower bound of expected maximum flow by increasing both the capacity and reliability of arcs.

3.9.1 Objective Function. The lower bound is the sum of expected flows for each path in the network. The expected flow through path j ($j = 1, 2, \ldots, q$) is $R_j f_j$, where R_j is a product of arc reliabilities

$$R_j = \prod_i (P_i + .1 \cdot X_i) \quad \text{for all arcs } i \ (i = 1, 2, \ldots, n) \text{ on path } j$$

and f_j is the flow through path j. The objective is maximize the lower bound by increasing the reliability, R_j, of each path. therefore the objective function is

$$\text{Max } \sum_{j=1}^{q} R_j f_j$$

The objective function is nonlinear since R_j is a product of X_i's.

3.9.2 Constraints. There are three types of constraints in the model. They are:

1. Constraints to ensure arc reliabilities are ≤ 1.
2. Flow capacity constraints for each path j.
3. A budget constraint indicating the amount of funds available for investment, and the cost of increasing capacity and reliability on each arc i.

The constraints to ensure reliabilities are less than 1 are

$$P_i + .1 \cdot X_i \leq 1$$

There is a reliability constraint for each arc in the network.

Let a_{ij} equal 1 if arc i lies on path j, and 0 otherwise. Using this definition, the constraints for flow capacity are

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i + d_i$$

Where u_i is the capacity of arc i, and d_i is the amount of capacity to add to arc i.

The budget constraint is

$$\sum_{i=1}^{n} (\alpha_{ci} d_i + \alpha_{ri} X_i) \leq \beta$$

Where α_{ci} is the cost of increasing the capacity of arc i by 1 unit, α_{ri} is the cost of increasing reliability of arc i by .1, and β is the total amount of the budget available to be invested.

3.9.3 Model. Combining the objective function and the constraints results in the following nonlinear model:

$$\text{Max } \sum_{j=1}^{q} R_j f_j \qquad \text{where } R_j = \prod_i (P_i + .1 \cdot X_i)$$

s.t.

$$P_i + .1 \cdot X_i \leq 1 \qquad i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i + d_i$$

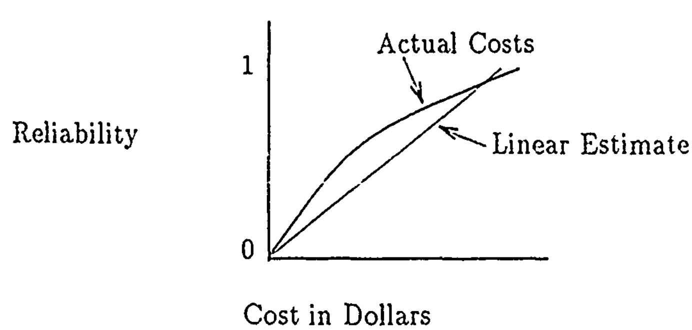
$$\sum_{i=1}^{n} (\alpha_{ci} d_i + \alpha_{ri} X_i) \leq \beta$$

$$X_i, P_i, f_j, \alpha_{ci}, \alpha_{ri}, \beta \geq 0 \text{ and } a_{ij} = 0, 1$$

The model will be referred to as Investment Strategy Model 7, and it's solution requires a nonlinear programming package.

3.10 Nonlinear Costs of Capacity and Reliability

The seven model formulations make an assumption that the cost of increasing capacity and reliability on an individual arc are linear. This is not always true in "real world" networks. It seems obvious that the cost to increase an individual arc's reliability from 0.9 to 1.0 would be higher than the cost to go from a reliability of say 0.4 to 0.5. The same goes for capacity. The cost to increase capacity from 4800 to 9600 would be more than going from 1200 to 2400 (capacity is sometimes increased in amounts that double the previous amount). If the reader wishes to use the actual costs (nonlinear curve), then he/she will have to fit a curve to the true reliability costs. This will complicate the models, but eliminates the bias of making the linear assumption. It should be stressed that the models are general, and may have to be modified to fit a specific application. A simple example of a reliability versus cost curve for an individual arc is seen below:



3.11 Summary

Seven different models to calculate investment strategies were formulated in this chapter. Investment Strategy Model 1 is a linear model that maximizes the lower bound by increasing the capacity of the arcs using continuous variables. Investment Strategy Model 2 is the same as Investment Strategy Model 1 except it requires an integer solution for the amount of capacity increase to an arc. Investment Strategy Model 3 is a nonlinear model that maximizes the lower bound by increasing the reliability of arcs. The variables are continuous in Investment Strategy Model 3. Investment Strategy Model 4 maximizes the upper bound by increasing reliability of arcs. It also uses continuous variables. Investment Strategy Models 5 and 6 use integer variables to optimize the lower and upper bounds respectively. Investment Strategy Model 5 is nonlinear, while Investment Strategy Model 6 is linear. Investment Strategy Model 7 optimizes the lower bound by increasing both capacity and reliability. Investment Strategy Model 7 uses continuous variables. Investment Strategy Models 1 and 2 were originally formulated by Yim (23), while Investment Strategy Models 3, 4, 5, 6, and 7 are new. The formulation of the models in this chapter completes research objective #1.

IV. Methodology

This chapter describes the methodology that was used in accomplishing the research objectives outlined in Chap' 'r I. The research objectives consist of formulating a set of models to determine the optimum investment strategy for improving reliability in a stochastic communications network, and then comparing the output from the models to another model which generates the optimum investment strategy for improving capacity. The following sections describe the steps this researcher used.

4.1 Understanding the Problem

This step was accomplished in the Literature Review presented in Chapter II. A comprehensive search was undertaken to find the most recent publications containing articles on stochastic communication networks. It was found that the performance of such networks depends on the topology, as well as the network parameters. The topology describes the general layout of the network, and can be in the form of a graph or a matrix. The reliability and capacity parameters describe the arcs and n' des. The interaction of topology and parameters determines the network's throughput and reliability. Network throughput is a function of both capacity and reliability, while the reliability of the network is merely a function of component reliability. The reader should not be under the impression that the calculation of throughput and reliability are a simple task, for they are not. Because the components can either operate or fail, the total number of states the network can be in is 2^n, where n is the number of components in the network. Even for a network with a small number of components, the number of states becomes quite large. For this reason, it is not easy to analyze stochastic communication networks. Because the number of states increases at an exponential rate, the calculation of

network throughput is considered an NP-class problem, which means that it will take an exponential amount of time to solve.

Since the throughput is a function of both capacity and reliability, then it seems obvious that the throughput could be increased if either or both were improved. The question comes up; which improvement strategy is better? The answer to this question is the primary reason this thesis was accomplished. DoD has recognized the dependency that the United States has on stochastic communications networks, and has asked the Air Force Institute of Technology to assist in the study of the performance through this type of network. Once the researcher had an understanding of stochastic communications networks, the next step was to formulate a set of models that described the performance of them.

4.2 Model Formulation

The model formulation step consisted of evaluating existing models and formulating new ones. Two models already existed to determine the optimum investment strategy for increasing arc capacity, and are presented in Chapter III. Both were developed by Yim in his thesis and are called Investment Strategy Models 1 and 2 (23). In addition, four new models were formulated to determine the optimum investment strategy when reliability is increased; they are called Investment Strategy Models 3, 4, 5, and 6. Investment Strategy Models 1, 2, 3, and 5 use the lower bound of expected maximum flow as the measure of network throughput. Investment Strategy Models 1 and 2 maximize the lower bound by increasing arc capacity, while Investment Strategy Model 3 uses continuous variables to maximize the lower bound by increasing arc reliability and evaluating potential arcs to see if they need to be added to the network. Once a potential arc has been added then it is a candidate for subsequent improvements in reliability. Investment Strategy Model 4 uses continuous variables to maximize the upper bound of expected maximum flow by improving arc reliability and adding potential arcs. Investment Strategy Model 5

is the same as Investment Strategy Model 3, except it requires an integer solution. Investment Strategy Model 6 is the same as Investment Strategy Model 4, except it requires an integer solution. Investment Strategy Models 1, 2, 4, and 6 are linear, whereas Investment Strategy Models 3 and 5 are nonlinear.

Another model presented in Chapter III, called Investment Strategy Model 7, was formulated to maximize the lower bound by increasing capacity and reliability simultaneously. It uses continuous variables. This model was used to perform sensitivity analysis on the test networks.

In addition to the seven investment models presented in Chapter III, there are two other models which are used in this research; both are presented in Chapter II. The first is a model that calculates the lower bound of expected maximum flow (see Section 2.5.1). Unlike the investment models discussed above, the lower bound model does not contain a budget constraint, and merely calculates the lower bound of the network without seeking to improve arc capacity or reliability. The second model calculates the upper bound of expected maximum flow without trying to improve capacity or reliability (see Section 2.5.2). The first research objective was accomplished by the model formulations presented in Chapter III.

4.3 Enhancements to Formula

As described in Section 2.6 of Chapter II, *Formula* is a Prolog-based program that formulates the lower and upper bound models, as well as Investment Strategy Models 1 and 2. It was written by Yim as a part of his thesis (23). To make the formulation of Investment Strategy Model 3 easier for large networks, *Formula* was enhanced to automatically formulate Investment Strategy Model 3 for input into *GINO*. *GINO* was then used to solve Investment Strategy Model 3. The enhanced version was called *Formula* Version 2.0, and is described in Chapter V.

4.4 *Design of Experiment*

After the models were formulated an experiment was designed to accomplish the second research objective. The experiment was run on five different sample networks. The sample networks ranged from a simple single path network to more complicated multipath networks (see Chapter VI for descriptions of the networks). The smaller networks were used because the reliability and throughput could be calculated manually. This provided a way to verify and validate that the models were working as designed. The following steps were accomplished on each network:

1. Calculate the lower bound, upper bound, expected throughput, expected reliability, and estimated reliability on the original network. These performance values provided a baseline in which to compare the investment strategies. The lower and upper bounds were found using the models described in Section 2.5 of Chapter II. *Formula* Version 2.0 was used to formulate the lower and upper bound models and LP/MIP-83 was used to solve them. LP/MIP- 83 is described in Chapter VII. The expected throughput and expected reliability of the original network were calculated by using the *Maxflo* simulation program described in Section 2.7 of Chapter 2. The estimated reliability was calculated using Equation 18 (see Chapter VIII for Equation 18).

2. Next, Investment Strategy Models 1 and 3 were run against the original network to determine the optimum investment strategy for increasing arc capacity and reliability. The output from Investment Strategy Model 1 calculated the new lower bound when arc capacity was increased, and identified which arcs should have their capacity increased. LP/MIP-83 was used to solve Investment Strategy Model 1. Investment Strategy Model 3 found the new lower bound by increasing arc reliability; in addition, it also identified which arcs should be added. *GINO* was used to solve Investment Strategy Model 3. Chapter VII contains a description of *GINO*.

3. The next step was to take the optimum investment strategies calculated in Step 2 and implement them into the original network. The investment strategies were implemented separately so a comparison could be made between the two models.
4. Once the investment strategies were implemented then the lower bound, upper bound, expected throughput, expected reliability, and estimated reliability were recalculated for each investment strategy. At this stage of the experiment the original performance values were known, as well as the performance values of the network after the investment strategies were implemented.

The experiment provided the steps necessary to find the optimum investment strategies, and generated the data so that an analysis could be made. Chapter VII contains a description of the experiment.

4.5 Analysis

An analysis was made on the data generated by the experiment, and the following questions were answered for each network:

1. Which investment strategy was better, and why?
2. Given the budget available, which arcs and nodes are the best candidates for reliability improvements?
3. Given the budget available, which arcs and nodes are the best candidates for capacity improvements?
4. What happens to the investment strategies when the cost of reliability is varied?

The analysis of the two investment strategies accomplished research objective #2. Chapter VIII contains the results and analysis from the experiment.

4.6 Use of Network Packages

A study was undertaken to determine the feasibility of solving the models described in Chapter III using network packages. Network packages are specifically designed to solve linear problems that can be formulated as a network using the node-arc incidence matrix. Chapter IX shows how the lower bound, upper bound, and Investment Strategy Models 1 and 4 can be reformulated and solved using a commercial package that supports networks with side constraints. The algorithms used by network packages are more efficient at solving network models than linear programming packages.

4.7 Application of Models in a Space Environment

The models and experiment outlined above can be used to analyze many types of stochastic communication networks. A study was undertaken to see how the methodology used on the sample networks would work if tried on a network that contained ground stations and earth orbiting satellites. Since DoD is relying more on space-based communications systems, it is obvious that the reliability and throughput of such systems is critical to the security of the United States. The study points out the options available for increasing capacity and reliability of a space-based communication system. It also describes many of the threats that space-based systems are exposed to, such as solar radiation and extreme temperature changes. The study is contained in Appendix D.

4.8 Summary

The methodology used in this chapter resulted in a set of models and an experiment to accomplish the research objectives outlined in Chapter I. The models have already be presented in Chapter III. To make the formulation of the models easier, the Prolog-based *Formula* program written by Yim was enhanced to formulate Investment Strategy Model 3. The enhanced *Formula* is called Version 2.0, and

is described in Chapter V. Chapter VII contains a description of the experiment, and the analysis from the experiment is in Chapter VIII. In addition, the linear models in Chapter III were reformulated so they could be solved using network packages. The results are presented in Chapter IX. Lastly, the methodology was applied to a space-based communication system (see Appendix D).

V. Formula Version 2.0

This chapter describes the computer program *Formula* Version 2.0. *Formula* Version 2.0 contains enhancements to *Formula* Version 1.0, which was originally written by Yim (23). As mentioned in Chapter II, *Formula* Version 1.0 is a computer program that formulates a set of models to analyze stochastic communication networks. It is written in the Prolog programming language. Without *Formula* Version 2.0, the formulation of the models described in Chapters II and III would be very cumbersome and labor intensive for large networks.

5.1 Features of Formula Version 1.0

Formula Version 1.0 will perform the following tasks:

1. Find all paths from the source to the sink, and calculate the reliability of each path.
2. Formulate the Maximum Flow Model.
3. Formulate the Lower Bound Model.
4. Formulate the Upper Bound Model.
5. Formulate Investment Strategy Model 1.
6. Formulate Investment Strategy Model 2.

The formulations generated in Tasks 2 thru 6 can be input directly into LP/MIP-83. LP/MIP-83 is a commercial off-the-shelf programming package that can solve integer and mixed integer programming problems. A brief description of LP/MIP-83 is contained in Chapter VII. *Formula* Version 1.0 uses a depth-first search to enumerate all the paths in the network (23:48-51). The reader should consult Yim's thesis for a detailed discussion of the depth-first search algorithm.

5.2 Enhancements to Version 1.0

Formula Version 1.0 was enhanced to formulate Investment Strategy Model 3 for direct input into *GINO*. *GINO* is a mathematical programming package capable of solving nonlinear programming problems. To formulate Investment Strategy Model 3, the format of the input file for *Formula* Version 1.0 had to be changed so the user could specify the cost of increasing reliability and the amount of the reliability budget. In addition, a set of subroutines had to be added to read the input file and formulate Investment Strategy Model 3. Also, an option allowing the user to request that Investment Strategy Model 3 be formulated had to be added to the main menu. Because of the extensive changes that were made to the original program, the name *Formula* Version 2.0 was given to the new program. A users manual for *Formula* Version 2.0 is contained in Appendix B, and a source code listing is in Appendix C.

5.3 Summary

This chapter describes the latest version of the *Formula* program, *Formula* Version 2.0. *Formula* Version 2.0 was written especially for the experiment described in Chapter VII. *Formula* Version 2.0 will enumerate all the paths in a network and compute their reliabilities. In addition, *Formula* Version 2.0 will formulate the following models: Maximum Flow, Lower Bound, Upper Bound, Investment Strategy Model 1, Investment Strategy Model 2, and Investment Strategy Model 3. All the models except Investment Strategy Model 3 are formulated for direct input into LP/MIP-83. Investment Strategy Model 3 is formulated for direct input into *GINO*. The users manual and source code listings for *Formula* Version 2.0 are contained in Appendices B and C respectively.

VI. Sample Networks

This chapter contains descriptions of the five sample networks that were used in this research. Each network was picked for a special reason which will become apparent as the networks are described. The networks are the same as those analyzed by Yim, Chan, and Marsh in their paper "Exact and Approximate Improvement to the Throughput of a Stochastic Network", and are called Networks 1, 2, 3, 4, and A. (24). Networks 1, 2, 3, and 4 are small networks, used to validate that the models formulated in Chapter III work correctly. Network A is a larger network that represents a more "realistic" scenario. A figure and brief description of each network is contained below.

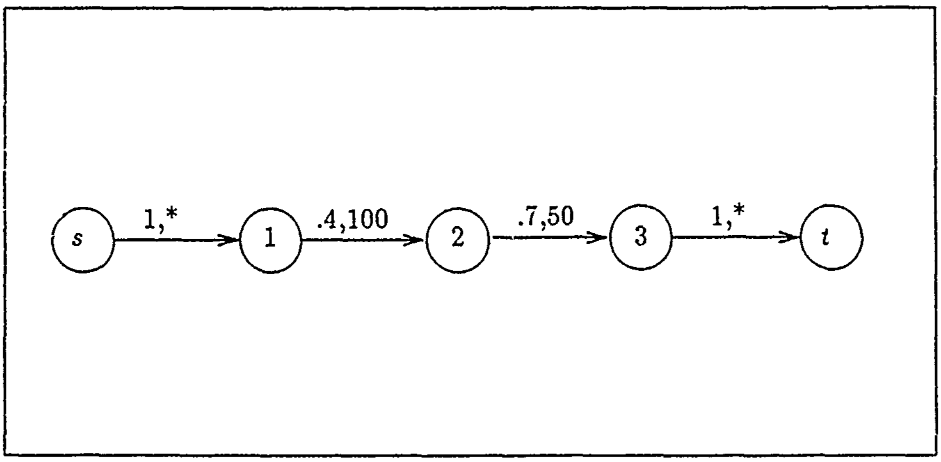


Figure 5. Network 1

6.1 Network 1

Network 1 has a single path from the source node, s, to the sink t (see Figure 5). This network was analyzed first because of its ease of calculation. The initial reliability of the network is simply the product of all the arc reliabilities, which

equals .28. The capacity bottleneck is the arc from Node 2 to Node 3 since it has the smallest capacity, i.e. 50. Network 1 does not have any potential arcs to be added. The nodes in Network 1 are not stochastic, i.e., they have a reliability of 1. In addition, the nodes have infinite capacity.

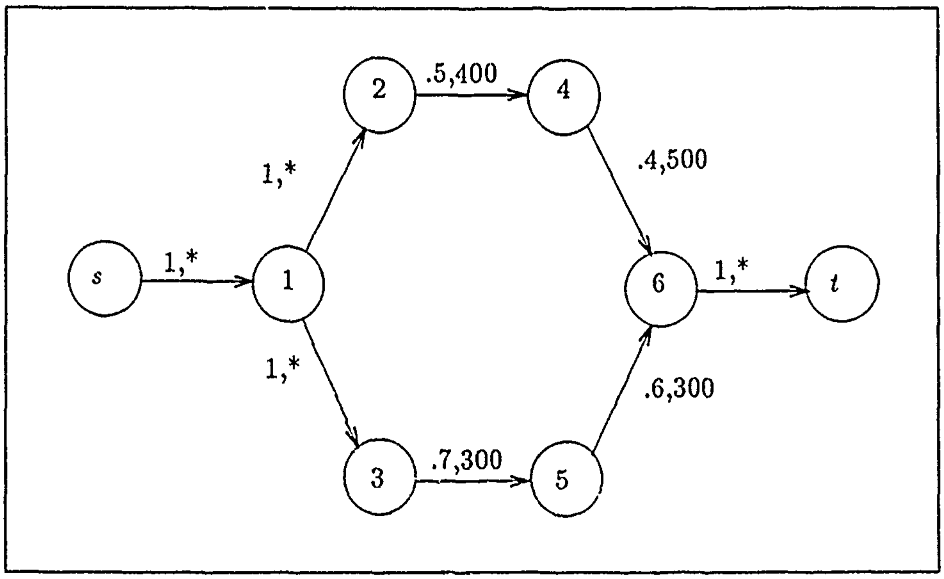


Figure 6. Network 2

6.2 Network 2

Network 2 has two paths from source to sink, as shown in Figure 6. This network is a little more complicated than Network 1, but the network reliability can still be calculated manually without much problem using a conventional series and parallel reliability equation as shown below:

$$\text{Network Reliability} = 1 * ((.5 * .4) + (.7 * .6) - (.5 * .4)(.7 * .6)) * 1 = 0.544$$

Each path has a capacity bottleneck. The upper path (Nodes s, 1, 2, 4, 6, and t) has its bottleneck on the arc between Nodes 2 and 4, while the lower path (Nodes s,

1, 3, 5, 6, and t) is constrained by the same amount, 300, by two different arcs. Just like Network 1, the nodes in Network 2 are not stochastic and have infinite capacity.

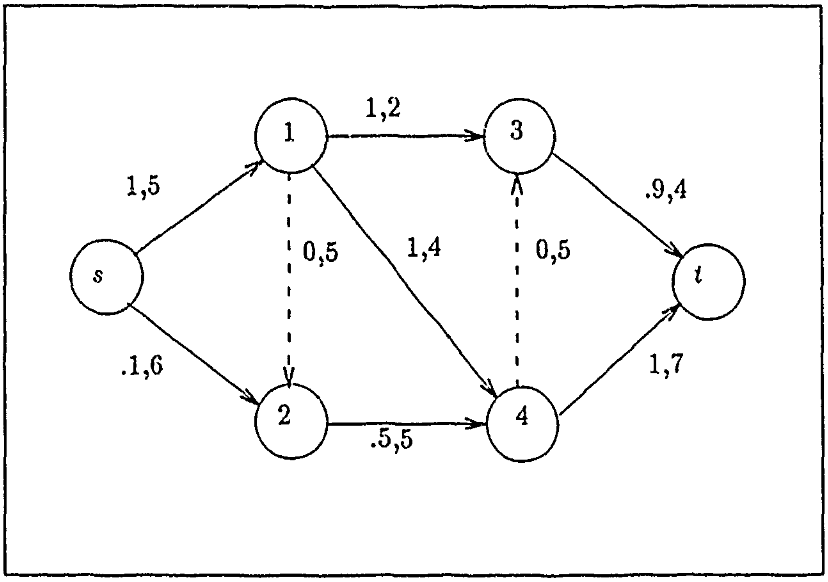
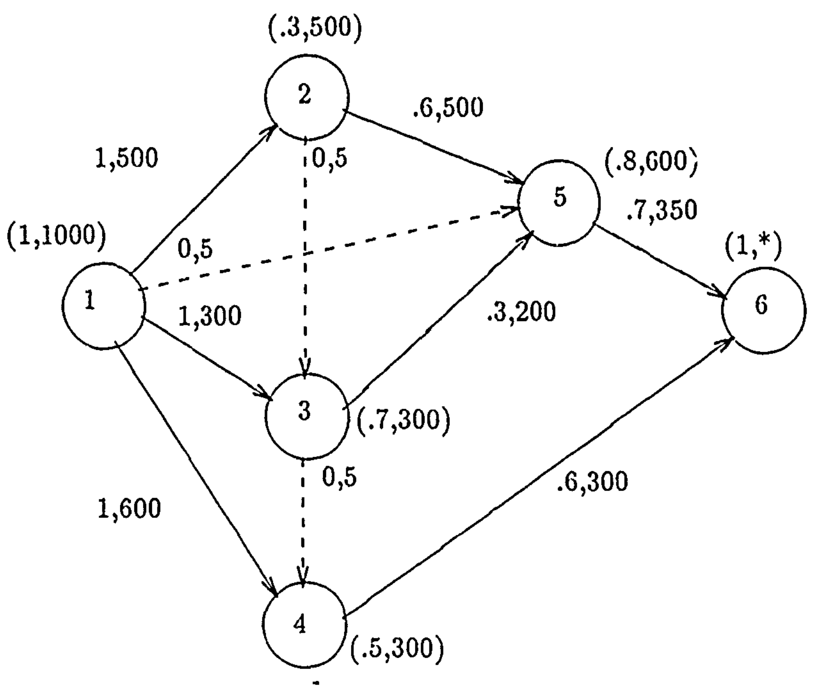


Figure 7. Network 3

6.3 Network 3

Network 3 can have a maximum of seven paths depending upon whether the potential arcs are added or not. Figure 7 shows a diagram of Network 3. The arcs represented by dotted lines are potential arcs which may be added to the network. If neither of the potential arcs is added then the number of paths is three. However, the number of paths increases to four if either of the potential arcs is added individually, and goes up to seven if both are added. Again the nodes in Network 3 are not stochastic and have infinite capacity. The reliability of Network 3 is not as easily calculated as the previous networks. For this reason, another method of calculating network reliability was used. The *Maxflo* simulation program, written by Bailey (4), uses Monte Carlo simulation to calculate the network reliability. *Maxflo* uses cutsets to determine if there is any flow going through the network. The arcs in the

cutset are either up or down depending on the draw from a random number generator. A network is considered to be 100 percent reliable, i.e., expected reliability = 1.0, if a path always exists from source to sink for every sample of the simulation. *Maxflo* will generate up to 100,000 samples.



KEY:
(X,Y) Reliability and Capacity of Node
X,Y Reliability and Capacity of Arc

Figure 8. Network 4

6.4 Network 4

Network 4 is the most complicated of the small networks. It has potential arcs and stochastic nodes. Figure 8 shows the network and provides a key on how to read the parameters. An artificial source node can be added, along with a dummy arc

connecting the artificial source node to Node 1; the dummy arc and source node will each have a reliability of 1 and an infinite capacity. A maximum of seven paths will exist if all three potential arcs are added. The stochastic nodes must be converted into arcs. Each arc has the same reliability and capacity as the stochastic node it is replacing. Two dummy nodes, each having a reliability of 1 and infinite capacity, are placed at each end of the arc that's replacing the node. The reader should consult Yim (23) for details on how to convert the stochastic nodes into arcs. Again *Maxflo* was used to calculate network reliability due to the size of the network.

6.5 Network A

Network A is shown in Figure 9. It is larger than the previous networks, and is more "realistic". Tables 1 and 2 show the capacity and reliability parameters of the arcs and nodes. In addition, Tables 3 and 4 show the dependent arc and node pairs. The dependent pairs represent a single medium. The medium is represented twice to prevent cycles in the network. If a component is dependent, then both of the arcs/nodes in the pair will fail if one fails. As seen in Figure 9, Network A has multiple sinks. Therefore, an artificial sink will have to be added to the network. The procedure for adding an artificial sink is described in the *Formula* Version 2.0 Users Manual contained in Appendix B.

6.6 Summary

Five sample networks were described. The sample networks represent a cross section of various topologies. The networks vary from a simple one path topology to more complicated multipath configurations. Chapter VII contains the description of the experiment used to determine whether increasing capacity or reliability is the better investment strategy for the sample networks.

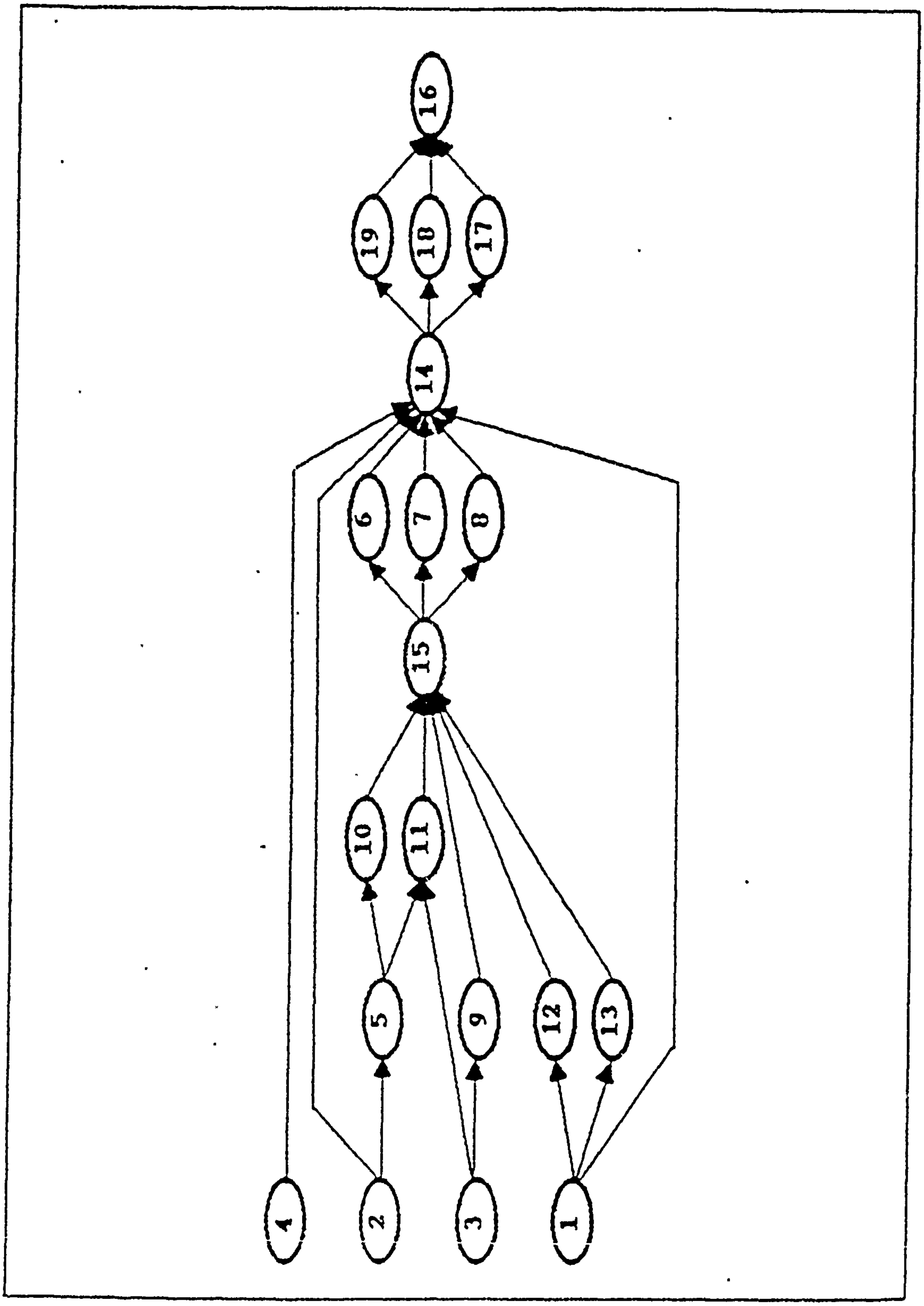


Figure 9. Network A (23:57)

Table 1. Description of Arcs in Network A

Start Node	Terminate Node	Reliability	Capacity
1	12	1.0	1200
1	13	1.0	1200
1	14	1.0	1200
2	5	0.3	1200
2	14	0.6	1200
3	9	1.0	1200
3	11	1.0	1200
4	14	1.0	1200
5	10	0.6	1200
5	11	0.7	1200
6	14	0.6	4800
7	14	0.6	4800
8	14	0.3	4800
9	15	1.0	4800
10	15	0.6	4800
11	15	1.0	4800
12	15	0.7	4800
13	15	1.0	4800
14	17	0.3	4800
14	18	0.6	4800
14	19	0.6	4800
15	6	0.3	4800
15	7	0.6	4800
15	8	0.7	4800
17	16	0.7	4800
18	16	0.6	4800
19	16	0.7	4800

Table 2. Description of Nodes in Network A

Node	Reliability	Capacity
1	1.0	*
2	0.3	*
3	0.7	*
4	0.5	*
5	0.8	*
6	1.0	*
7	0.3	*
8	0.7	*
9	0.5	*
10	0.8	*
11	1.0	*
12	0.3	*
13	0.7	*
14	0.5	*
15	0.8	*
16	0.8	*
17	0.7	*
18	0.3	*
19	1.0	*

* implies capacity is infinite

Table 3. Dependent Arcs in Network A

Dependent Pairs			
Start Node	Terminate Node	Start Node	Terminate Node
6	14	14	19
7	14	14	18
8	14	14	17
15	6	19	16
15	7	18	16
15	8	17	16

Table 4. Dependent Nodes in Network A

Dependent Pairs	
Node	Node
6	19
7	18
8	17
15	16

VII. *Design of Experiment*

This chapter contains the description of an experiment that uses three of the models formulated in Chapter III. The experiment used the models to determine the optimum investment strategy for increasing capacity and reliability in the five sample networks described in Chapter VI. First, Yim's Investment Strategy Model 1 was used to determine the optimum investment strategy for increasing arc capacity. Then Investment Strategy Model 3 was run on the same networks to find the best investment strategy for improving reliability and adding arcs. Lastly, Investment Strategy Model 7 was used to determine how sensitive the lower bound was to changes in the cost of capacity and reliability. The experiment could be enhanced to include additional models that were formulated in Chapter III. For example, if the application required that the capacity or reliability be increased in discrete amounts, then Investment Strategy Models 2 and 5 could be used instead of Investment Strategy Models 1 and 3. The following sections describe the assumptions, tools, files, and process steps used in the experiment.

7.1 Assumptions

The following assumptions were made:

1. The costs of increasing capacity by 1 unit or reliability by .1 are the same.
2. There is only one budget available, and it is used for either increasing capacity or reliability.
3. The flow capacity of potential arcs is set to 5. This value is picked for convenience, but any positive value can be used.

The first assumption had to be made because actual cost data for increasing reliability was not available. The cost data for increasing capacity came from Yim, Chan,

and Marsh (24). Since this assumption greatly impacts which investment strategy is better, Investment Strategy Model 7 was used to determine how sensitive the networks were to changes in the reliability cost. The second assumption implies that the budget cannot be split up into two parts. This means that any surplus cannot be shifted to the other investment strategy. There will never be any surplus when Investment Strategy Model 1 is used because there is no limit on how much capacity can be added to an arc. However, if Investment Strategy Model 3 is being used then all purchases of reliability will stop when the reliability of each arc equals one; this would result in a surplus.

7.2 Tools

The experiment required the following software packages: *Arity/Prolog* Version 5.0 Interpreter, *Formula* Version 2.0, *Maxflo*, LP/MIP-83, and *GINO*. In addition, the following computers were used: a Sun 386i workstation, a Digital Equipment Corporation (DEC) 11/785 miniframe, and a DEC 8550 mainframe. A description of each tool is described below.

7.2.1 Arity/Prolog Interpreter. The *Arity/Prolog* Version 5.0 Interpreter is an enhanced Prolog interpreter written by the Arity Corporation of Concord, Massachusetts (3). It supports the standard Prolog convention, plus some enhanced features such as screen manipulation. The interpreter will operate on any IBM-compatible microcomputer and was used to run the *Formula* Version 2.0 program.

7.2.2 Formula Version 2.0. As described in Chapter V, *Formula* Version 2.0 is a program that formulates both linear and nonlinear models for stochastic communication networks. *Formula* Version 2.0 was used to formulate the following models:

1. Lower Bound
2. Upper Bound
3. Investment Strategy Model 1
4. Investment Strategy Model 3

A users guide for *Formula* Version 2.0 is contained in Appendix B.

7.2.3 Maxflo. *Maxflo* is a Monte Carlo simulation program written by Bailey (4) to calculate the expected maximum flow and expected reliability for a stochastic communication network. The simulation program is written in FORTRAN 77 and has been run successfully using both Unix and VMS operating systems. A brief description of *Maxflo* was given in Chapter II, but the reader should consult Bailey (4) for a detailed description of the program.

7.2.4 LP/MIP-83. LP/MIP-83 is used to solve linear programming problems. The user must provide an objective function, either minimize or maximize, and the appropriate constraints. LP/MIP-83 can solve both continuous and mixed integer problems. The program is written by Sunset Software of San Marino, California and runs on any IBM-compatible microcomputer (21). LP/MIP-83 was used to solve the following models: Lower Bound, Upper Bound, and Investment Strategy Model 1.

7.2.5 GINO. *GINO.* General INteractive Optimizer, is a command oriented computer program for solving optimization problems. The program was written by Lindo Systems Inc. of Chicago, Illinois, and can be used to solve nonlinear problems. The user must specify an objective function and whether the objective function is to be maximized or minimized. In addition, the user has to input all of the constraints, to include non-negativity. One important fact to stress about *GINO* is that the

solution found may not be the global optimal, only a local (15:28). *GINO* was used to solve Investment Strategy Model 3.

7.2.6 Computer Equipment. The Sun 386i workstation is a hybrid system that runs both the MS-DOS and Unix operating systems. The 386i workstation was used to run the *Arity/Prolog* Interpreter, *Formula* Version 2.0, and LP/MIP-83 programs. When the 386i workstation was running these programs, it was using the MS-DOS operating system. The DEC 11/785 miniframe used in the experiment was running Unix Version 4.3 (Berkley System Domain). *Maxflo* was run on the DEC 11/785. The DEC 8550 was running VMS Version 5.3. VMS is a proprietary operating system of DEC. *GINO* was run on the DEC 8550. It should be stressed that the following experiment could be performed on any computer arrangement that supports the MS-DOS, Unix, and VMS operating systems.

7.3 Sample Networks

The networks used in the experiment are described in Chapter VI. They represent a mix of various topologies. The experiment was run on each network to determine the best investment strategy.

7.4 Files.

There are ten different files used in the experiment. To simplify the description of the files, the following notation will be used:

- The term *converted* network will be used to denote a network that has been converted from nodes and arcs to one which contains strictly arcs. The procedure for converting a network is described in Section 2.6 of Chapter II.
- The term *modified* network will be used to denote a *converted* network that has been modified by the appropriate investment strategy.

As stated above, there are ten different files used in the experiment. A brief description of each follows:

1. *Convert.for* is a file used by *Formula* which contains the *converted* network.
2. *Modify1.for* is a file used by *Formula* which contains the *modified* network after investment strategy 1, increased capacity, has been implemented.
3. *Modify3.for* is a file used by *Formula* which contains the *modified* network after investment strategy 3, increased reliability, has been implemented.
4. *Convert.max* is a file used by *Maxflo* which contains the *converted* network.
5. *Modify1.max* is a file used by *Maxflo* which contains the *modified* network after investment strategy 1, increased capacity, has been implemented.
6. *Modify3.max* is a file used by *Maxflo* which contains the *modified* network after investment strategy 3, increased reliability, has been implemented.
7. *output3.lp* is a file generated by *Formula* which contains the formulation for the Lower Bound model.
8. *output4.lp* is a file generated by *Formula* which contains the formulation for the Upper Bound model.
9. *output5.lp* is the file generated by *Formula* which contains the formulation for Investment Strategy Model 1.
10. *output7.nlp* is the file generated by *Formula* which contains the formulation for Investment Strategy Model 3.

7.5 Actual Experiment

This section describes the steps that were accomplished in the experiment. The experiment was performed on each of the five networks. Figure 10 shows an overview of the experiment.

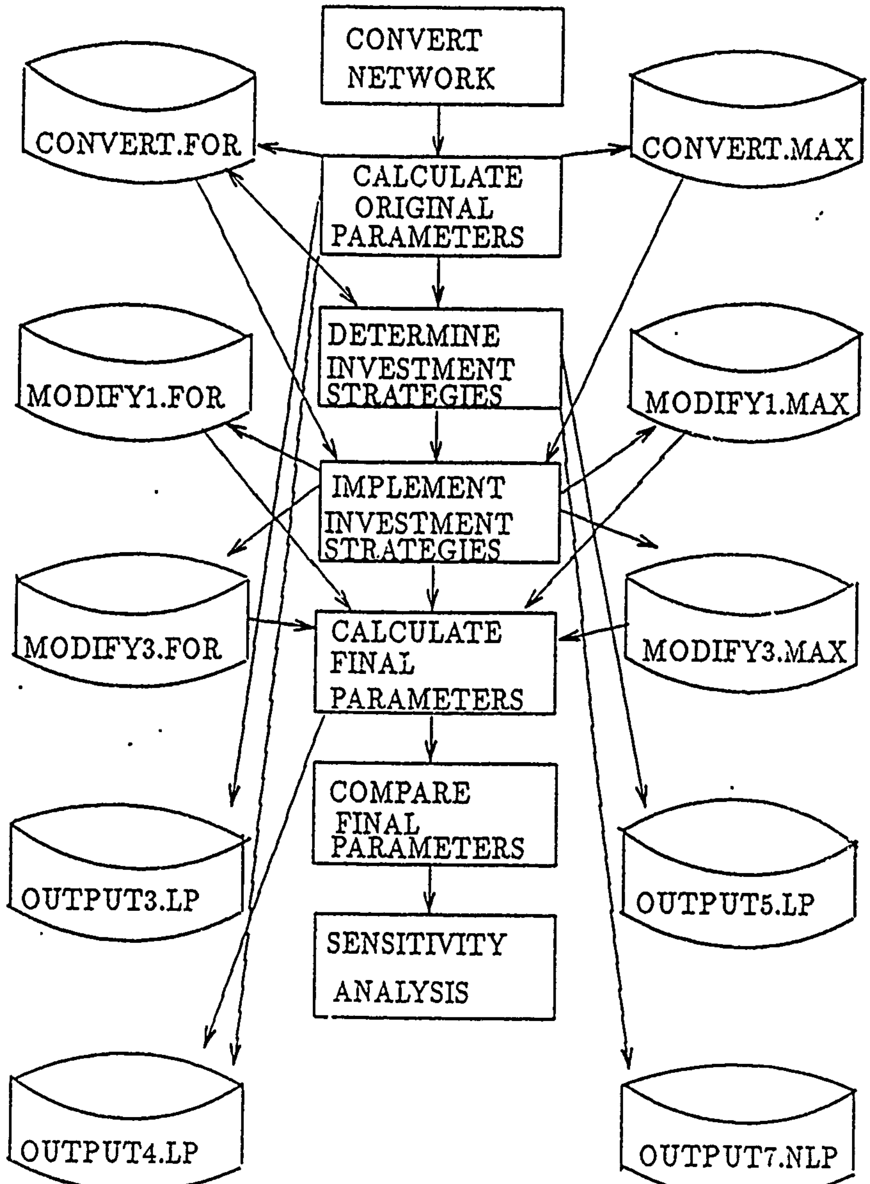


Figure 10. Diagram of Experiment

7.5.1 Convert Network. The original network was converted into a network consisting of a single source and single sink. In addition, the stochastic nodes were converted to arcs connected by dummy nodes. The resulting network is called a *converted* network for the purpose of distinguishing it from the original. It should be stressed that the *converted* network is equivalent to the original network. The procedure for converting the network is contained in the *Formula* Version 2.0 User's Manual (see Appendix B).

7.5.2 Calculate Original Parameters for the Network. The following parameters were calculated to establish a baseline in which to compare the *converted* network to the *modified* network:

1. Lower Bound
2. Upper Bound
3. Expected Throughput
4. Expected Reliability

The methods for calculating each parameter is described below.

7.5.2.1 Calculation of Original Lower and Upper Bounds.

1. A description of the *converted* network was entered into a *Formula* input file using a standard wordprocessor, such as Wordperfect or Wordstar, on the 386i workstation. The *Formula* Version User's Manual in Appendix B describes how to create the input file. The file was saved as *Convert.for.*
2. The *Arity/Prolog* interpreter was then used to start *Formula* Version 2.0 (see Appendix B for details).
3. Once *Formula* was started, *Convert.for* was used as the input file and the "Generate the Lower Bound Formulation" option was selected from the main

menu. *Formula* wrote the lower bound formulation to an output file called *output3.lp*.

4. Next, the "Generate the Upper Bound Formulation" option was selected from the main menu. *Formula* wrote the upper bound formulation to an output file called *output4.lp*.

5. After *Formula* finished generating the models in Steps 3 and 4, then the *Formula* and *Arity/Prolog* interpreter programs were terminated.

6. At this time the LP/MIP-83 program was started and it solved the models contained in the *output3.lp* and *output4.lp* files generated by steps 3 and 4. The output reports from LP/MIP-83 contained the original lower and upper bounds for the *converted* network.

7.5.2.2 Calculation of Original Throughput and Reliability.

1. The *Maxflo* program was started on the DEC 11/785.

2. A description of the *converted* network, created in section 7.5.1, was entered into *Maxflo* (see Bailey (4) for details on how to enter the network).

3. After the *converted* network had been entered, then the *Maxflo* simulation was run for 10,000 samples. The results represented the original expected throughput and reliability of the *converted* network.

4. At this time a file containing a description of the *converted* network was saved as *Convert.max* using the save option from *Maxflo's* main menu.

7.5.3 Determine the Optimum Investment Strategies. Two investment strategies were calculated on the *converted* network. First, the Investment Strategy Model 1 was used to determine the optimum investment strategy for increasing capacity on the arcs. Next, Investment Strategy Model 3 was solved to find the optimum investment strategy for increasing reliability in existing arcs and by adding potential

arcs. The detailed procedures for calculating the two investment strategies were accomplished using the 386i workstation and DEC 8550. The procedures are described below:

1. The budgets and costs values for arc capacity and reliability were entered into the *Convert.for* file using a standard wordprocessor (see Appendix B for details) on the 386i. The file was then saved (the file is still called *Convert.for*).
2. The *Arity/Prolog* interpreter was used to start *Formula* Version 2.0.
3. Once *Formula* was started, then *Convert.for* was used as the input file and the "Generate the Investment Strategy Model 1" option was selected from the main menu. *Formula* then formulated the model and wrote it to an output file called *output5.lp*.
4. Next, the "Generate the Investment Strategy Model 3" option was selected from the main menu. The model resulting from this formulation was written to an output file called *output7.nlp*.
5. The *Formula* and *Arity/Prolog* interpreter programs were then terminated.
6. At this time the LP/MIP-83 program was started and it solved Investment Strategy Model 1 contained in the *output5.lp* file generated by Step 3. The output report of LP/MIP-83 contained the optimal solution to Investment Strategy Model 1. The LP/MIP-83 report included a new lower bound and the amount of capacity increase for each arc.
7. The LP/MIP-83 program was then terminated.
8. Next, the *GINO* program was started on the DEC 8550 and it solved Investment Strategy Model 3 contained in the *output7.nlp* file generated by Step 4. The output report from *GINO* contained the optimal solution to Investment Strategy Model 3, which is the new lower bound that occurs when arc reliabilities are increased and potential arcs are added. In addition to the new lower

bound, the output report from *GINO* will contain a list of the X_i variables and their values for the optimal solution. The X_i variables will represent how much the reliability should be improved on each arc.

9. The last step was to terminate the *GINO* program.

7.5.4 Implement the Optimum Investment Strategies. The investment strategies found in Section 7.5.3 were then implemented into the *converted* network using the following processes.

7.5.4.1 Implement the Increased Capacity Strategy. The following steps were used to implement the results from Investment Strategy Model 1:

1. Using the 386i workstation, a copy was made of the *converted* network file, *Convert.for*; the copy was called *Modify1.for*.
2. At this time the *Modify1.for* file, created in Step 1, was updated using a standard wordprocessor to include the new arc capacities that were generated by the LP/MIP-83 report in Step 6 of Section 7.5.3.
3. The updated *Modify1.for* file was then saved (the file was still called *Modify1.for*).
4. The *Maxflo* program was started on the DEC 11/785 miniframe.
5. At this time the *Convert.max* file, created in Step 4 of Section 7.5.2.2, was retrieved using the retrieve option from *Maxflo's* main menu.
6. Once the *Convert.max* file had been retrieved, then it was updated using the new capacities generated by LP/MIP-83 in Step 6 of Section 7.5.3. The resulting network is a *modified* network. The *modified* network was then saved in the *Modify1.max* file.

7.5.4.2 Implement the Increased Reliability Strategy. The following steps were used to implement the results from Investment Strategy Model 3:

1. Using the 386i workstation, a copy was made of the *converted* network file, *Convert.for*; the copy was called *Modify3.for*.

2. At this time the *Modify3.for* file, created in Step 1 above, was updated using a standard wordprocessor to include the new arc reliabilities that were generated by *GINO* in Step 8 of Section 7.5.3.

3. The updated *Modify3.for* file was then saved (the file was still called *Modify3.for*).

4. The *Maxflo* program was started on the DEC 11/785 miniframe.

5. At this time the *Convert.max* file, created in Step 4 of Section 7.5.2.2, was retrieved using the retrieve option from *Maxflo's* main menu.

6. Once *Convert.max* had been retrieved, then the *converted* network was modified using the new reliabilities generated by *GINO* in Step 8 of Section 7.4.3. The resulting network is a *modified* network. The *modified* network was then saved in the *Modify3.max* file using the save feature of *Maxflo*.

7.5.5 Calculate Final Parameters for the Network. This section explains the process for determining the final parameters of the *modified* network. Investment Strategy Models 1 and 3 calculated the optimum lower bounds for the two separate investment strategies (see Section 7.5.3). Therefore, the only remaining parameters that needed to be calculated were the upper bound, expected throughput, and expected reliability of the *modified* network. The calculation of the final parameters used the *Modify1.for*, *Modify3.for*, *Modify1.max*, and *Modify3.max* files. The following sections describe how the final upper bound, expected throughput, and network reliability parameters were calculated for both investment strategies.

7.5.5.1 Calculate the Final Parameters for the Increased Capacity Strategy. The final upper bound, expected throughput, and network reliability that resulted from increasing arc capacity were calculated using the following steps:

1. Using the 386i workstation, the *Arity/Prolog* interpreter was used to start *Formula*. *Modify1.for* was used as the *Formula* input file.

2. Next, the "Generate the Upper Bound Formulation" was selected from the main menu. *Formula* wrote the upper bound formulation to *output4.lp* (replacing the old version of *output4.lp* created in Step 4 of Section 7.5.2.1).

3. After *Formula* finished generating the upper bound model in Step 2 above, then the *Formula* and *Arity/Prolog* interpreter programs were terminated.

4. At this time the LP/MIP-83 program was started and it solved the upper bound model contained in *output4.lp*. The output report from LP/MIP-83 contained the final upper bound when Investment Strategy Model 1 was used.

5. The *Maxflo* program was started on the DEC 11/785 miniframe, and the *Modify1.max* file was retrieved.

6. After *Modify1.max* had been retrieved, the *Maxflo* simulation was run for 10,000 samples to calculate the expected throughput and expected reliability of the *modified* network. The results contain the final expected throughput and expected reliability of the *modified* network when Investment Strategy Model 1 was used.

7.5.5.2 Calculate the Final Parameters for the Increased Reliability Strategy. The final upper bound, expected throughput, and network reliability that resulted from increasing arc reliability were calculated using the following steps:

1. Using the 386i workstation, the *Arity/Prolog* interpreter was used to start *Formula*. *Modify3.for* was used as the *Formula* input file.

2. Next, the "Generate the Upper Bound Formulation" was selected from the main menu *Formula* wrote the upper bound formulation to *output4.lp* (replacing the old version of *output4.lp* created in Step 2 of Section 7.5.5.1).

3. After *Formula* finished generating the upper bound model in Step 2 above, then the *Formula* and *Arity/Prolog* interpreter programs were terminated.

4. At this time the LP/MIP-83 program was started and it solved the upper bound model contained in *output4.lp*. The output report from LP/MIP-83 contained the final upper bound when Investment Strategy Model 3 was used.

5. The *Maxflo* program was started on the DEC 11/785 miniframe, and *modify3.max* was retrieved.

6. After *Modify3.max* had been retrieved, the *Maxflo* simulation was run for 10,000 samples to calculate the expected throughput and expected reliability of the *modified* network. The results contain the final expected throughput and expected reliability of the *modified* network when Investment Strategy Model 3 was used.

7.5.6 Compare Final Parameter Values. At this point in the experiment, both the original and final parameters have been calculated. The original parameters represented the lower bound, upper bound, expected throughput, and expected reliability of the *converted* network, which is equivalent the *original* network. On the other hand, the final parameters represent the same critical values on the *modified* network after the optimum investment strategies have been implemented. Chapter VIII contains the results and analysis of the four sample networks.

7.5.7 Sensitivity Analysis. Investment Strategy Model 7 was used to perform a sensitivity analysis to determine the impacts to the investment strategies when the cost of reliability was varied. To do this, the budget and cost of capacity were held constant while the cost of reliability was varied.

7.6 Summary

The experiment described in this chapter was used to analyze two different investment strategies, and their impacts on a set of sample stochastic communication networks. Investment Strategy Model 1 was used to determine the optimum investment strategy for increasing capacity, while Investment Strategy Model 3 was used to find the optimum investment strategy for increasing reliability. Investment Strategy Model 7 was used for sensitivity analysis to determine the effects of changing the cost of reliability. Also included is a description of the software, computer equipment, and processing steps required to do the experiment. When the experiment is performed on a network, the following parameters are calculated for both the original network and the new networks that result from implementing the investment strategies:

1. Lower Bound
2. Upper Bound
3. Expected Throughput
4. Expected Reliability

The experiment was performed on all five networks described in Chapter VI. The results and analysis of the outcomes are contained in Chapter VIII.

VIII. *Results and Analysis*

This chapter contains the results and analysis of increasing capacity and reliability in the five sample networks described in Chapter VI. The experiment took each stochastic communication network and first determined the optimum investment strategy for increasing capacity using Investment Strategy Model 1. Next, the optimum investment strategy for increasing reliability was calculated using Investment Strategy Model 3. The last step was to perform a sensitivity analysis using Investment Strategy Model 7. The purpose of the sensitivity analysis was to determine the impacts on the investment strategies when the cost of reliability was varied. The following sections explain the results of the experiment for each of the five sample networks.

The lower bound was used as the measurement of network throughput for both investment strategies (see model formulations in Chapter III for details). As will be shown later, the lower bound is a good approximator of the expected throughput in a stochastic communication network. A Monte Carlo simulation program called *Maxflo* was used to calculate the expected throughput. *Maxflo* will also calculate the expected reliability. The *Maxflo* program was run three times on each sample network. First, it was run against the original networks as they are shown in the Figures 5, 6, 7, 8, and 9 of Chapter VI. Then *Maxflo* was run again after each investment strategy was implemented. By calculating the expected throughput before and after the investment strategies were implemented, it was possible to compare the investment strategies against each other. A description of the experiment that determined the better investment strategy is contained in Chapter VII.

The following parameters were calculated on the original network and the networks that resulted after the investment strategies were implemented:

1. Lower Bound
2. Upper Bound
3. Expected Throughput
4. Expected Reliability

A new parameter introduced in this chapter is the estimated reliability of a network. The equation for calculating estimated reliability comes from the "Exact and Approximate Improvement to the Throughput of a Stochastic Network" paper prepared by Yim, Chan, and Marsh (24:12). They defined network reliability to be:

$$R = 1 - \frac{\sum_{j=1}^{r^*} R_j}{\sum_{j=1}^{r\prime} R_j} \tag{18}$$

where

$$\begin{aligned} R_j &= \text{the reliability of path } j \\ r\prime &= \text{the total number of paths in the network} \\ r^* &= \text{the number of paths actually in use} \end{aligned}$$

Yim, Chan, and Marsh admit that this is a simple indicator, and works better on large networks.

The following analysis was made based on the assumptions and restrictions outlined in Chapters I, III, and VII. Appendix A contains the data generated by the experiment described in Chapter VII.

8.1 Analysis of Network 1

As shown in Figure 5, Network 1 is a single path network. Tables 5, 6, and 7 contain the results of the experiment for Network 1. The original and final parameters are contained in Table 5. The original parameters represent the lower bound,

Table 5. Original and Final Parameters of Network 1

Network	Lower Bound	Expected Throughput	Upper Bound	Expected Reliability	Estimated Reliability
Org	14.0	14.04	35.0	0.2808	0
ISM1	19.6	19.06	40.0	0.2808	0
ISM3	50.0	50.00	50.0	1.0000	0

Notes:
Org is the original network
ISM1 is the network after Investment Strategy Model 1
ISM3 is the network after Investment Strategy Model 3

upper bound, expected throughput, expected reliability, and estimated reliability of the original network, while the final parameters contain the values for the network after Investment Strategy Models 1 and 3 were implemented. As shown in Table 5, investing in reliability is clearly the better strategy. The budget available for investment was $100, and the cost of increasing capacity by 1 or reliability by .1 was $5. A close study of Table 5 reveals that the lower bound is a much better estimator of the expected throughput for this simple network. Also the reader should note that the estimated reliability value is not even close to the expected reliability calculated by the Monte Carlo simulation. Since the nodes in Network 1 were not stochastic, they were not considered for potential investments.

Which investment strategy is better, and why?

As indicated above, the better strategy is to invest in reliability. The data in Table 5 clearly indicates this fact. All the final parameters for Investment Strategy Model 3 are greater than those of Investment Strategy Model 1. While it is not shown in the tables, the amount of budget spent by Investment Strategy Model 3 was only $45, where as Investment Strategy Model 1 used the entire $100 (see Appendix A.1 for the actual models and results). The fact that the entire budget was spent for Investment Strategy Model 1 is not surprising since there is no upper

Table 6. Reliability Improvements For Arcs In Network 1

Start Node	Terminate Node	Initial Reliability	Reliability Increase	Final Reliability
1	2	0.4	0.6	1.0
2	3	0.7	0.3	1.0

limit on the amount of capacity that can be added to an existing arc. However, the highest reliability that can be achieved by an arc is 1. Because of this, once all arcs have reached a reliability of 1 in Investment Strategy Model 3 there is no need to spend any more money on improving the network. Another important point to note is that increasing reliability will improve all the final parameters, but increasing capacity will have no effect on reliability. When the network was made fully reliable, i.e., all arcs in the network have a reliability of 1, then the lower and upper bounds were equal. This is because the network is no longer stochastic. Since the network is no longer stochastic, the expected throughput is actually the maximum flow through the network.

Given the budget available, which arcs are the best candidates for reliability improvements?

Table 6 shows the arcs that were improved by Investment Strategy Model 3. Only two arcs were chosen for improvement. This is because the rest of the arcs already had a reliability of 1. Both arcs were improved until their reliabilities equaled 1. Once the reliabilities reached 1 then there was no need to continue spending money because all components were at the limit for reliability. As a result, only \$45 was spent out of the budget of \$100. The reader should also notice from Table 5 that the expected reliability of the network is 1 after Investment Strategy Model 3 is implemented. This implies that there is always a path between the source and sink node. This point is trivial for Network 1 which has only one path, but will become

Table 7. Capacity Improvements For Arcs In Network 1

Start Node	Terminate Node	Initial Capacity	Capacity Increase	Final Capacity
2	3	50	20	70

more important in the analysis of the more complicated networks. The estimated reliability calculated by Equation 18 is equal to zero because there is only one path in the network. This indicates that Equation 18 is not a good estimator of network reliability when the network has just one path.

Given the budget available, which arcs are the best candidates for capacity improvements?

Table 7 contains only 1 arc, from Node 2 to Node 3. This arc received 20 units of improvement, raising the capacity from 50 to 70 units, thereby using up the entire budget of \$100. The reason this arc received all the increase was because the only other arc which was a candidate for capacity increase was from Node 1 to Node 2, and it already had a capacity of 100.

What happens to the investment strategies when the cost of reliability is varied?

By leaving the cost of capacity fixed at \$5 and the budget constant at \$100 in Investment Strategy Model 7, it was discovered that increasing reliability was the better investment strategy until the cost of reliability reached \$90. When the cost of reliability exceeded \$90, then increasing capacity became the better strategy. Therefore, as long as the price of reliability is \$90 dollars or less then the user should invest in reliability. The reason that the price of reliability had to increase over 18 fold before capacity became the better investment strategy is that the average path reliability of Network 1 is only 0.28.

Table 8. Original and Final Parameters of Network 2

Network	Lower Bound	Expected Throughput	Upper Bound	Expected Reliability	Estimated Reliability
Org	206.0	209.55	380.0	0.5402	0
ISM1	210.2	213.83	386.0	0.5402	0
ISM3	700.0	700.00	700.0	1.0000	0

Notes:
Org is the original network
ISM1 is the network after Investment Strategy Model 1
ISM3 is the network after Investment Strategy Model 3

8.2 Analysis of Network 2

As shown in Figure 6, Network 2 has 2 paths. Both paths have arcs which are candidates for improvements in capacity and reliability. The nodes in Network 2 are not stochastic, therefore they will not be candidates for investment. Just as in Network 1, the better investment strategy is to improve reliability instead of capacity (see Table 8). The budget for improving Network 2 was set at \$1000, while the cost of increasing capacity by 1 or reliability by .1 was \$50. The data in Table 8 shows that the lower bound is a better estimator of the expected throughput than the upper bound. Again the estimated reliability is zero because there are only two paths, and they both are in use. Tables 8, 9, and 10 contain the results of the investment models on Network 2. The reports for Network 2 are in Appendix A.2.

Which investment strategy is better, and why?

As shown in Table 8, the better strategy is to improve reliability. By improving reliability, the lower bound was increased by over a factor of 3, from 210.2 to 700.0, while increasing capacity only improved the lower bound by a total of 4.2 units. The increased capacity strategy used the entire budget as expected, but the increased reliability strategy only used \$900, leaving a surplus of \$100. The surplus

Table 9. Reliability Improvements For Arcs In Network 2

Start Node	Terminate Node	Initial Reliability	Reliability Increase	Final Reliability
2	4	0.5	0.5	1.0
4	6	0.4	0.6	1.0
3	5	0.7	0.3	1.0
5	6	0.6	0.4	1.0

was achieved because all arcs had reached a reliability of 1 and there was no need to spend additional funds. Just like in Network 1, the lower and upper bounds are equal when the strategy to increase reliability is used. Again this is because all arcs have a reliability of 1 in the optimal solution, and the network is no longer stochastic.

Given the budget available, which arcs are the best candidates for reliability improvements?

Table 9 shows the arcs which had their reliabilities increased. Because the budget was large enough, the reliability of each arc was increased until it's reliability reached 1. The other arcs were not improved because their reliabilities were already equal to 1 (see Figure 6). This resulted in a $100 surplus.

Given the budget available, which arcs are the best candidates for capacity improvements?

Table 10. Capacity Improvements For Arcs In Network 2

Start Node	Terminate Node	Initial Capacity	Capacity Increase	Final Capacity
3	5	300	10	310
5	6	300	10	310

As shown in Table 10, only two arcs had their capacity increased. Both were increased by 10 units. This occurred because both had the same initial capacity of 300. Since the flow through a path is limited by the smallest capacity along that path, it would have been useless to increase the capacity of only one of the arcs. If one arc had been increased by the entire 20 units then the smallest capacity would still have been 300 and the improvement would have been for naught. As stated above, the entire budget was expended when Investment Strategy Model 1 was used.

What happens to the investment strategies when the cost of reliability is varied?

By leaving the cost of capacity fixed at \$50 and the budget constant at \$1000 in Investment Strategy Model 7, it was discovered that increasing reliability was the better investment strategy until the cost of reliability reached \$5150. When the cost of reliability exceeded \$5150, then increasing capacity became the better strategy. Therefore, as long as the price of reliability is \$5150 dollars or less then the user should invest in reliability. Just like in Network 1, the reason reliability was the better investment is because the average path reliability of Network 2 is very low, 0.31.

8.3 Analysis of Network 3

Network 3 consists of both existing and potential arcs as shown in Figure 7. By adding the potential arcs, the number of paths can increase from three up to a maximum of seven. The total budget was \$100 and the cost of increasing capacity by 1 or reliability by .1 was \$5. Just like the two previous networks, the lower bound was a better estimator of expected throughput than the upper bound, and the estimated reliability value is equal to zero because all available paths are in use. The results for Network 3 are contained in Tables 11, 12, and 13. The nodes in Network 3 were not candidates for improvement since they are not stochastic. The reports for Network 3 are in Appendix A.3.

Table 11. Original and Final Parameters of Network 3

Network	Lower Bound	Expected Throughput	Upper Bound	Expected Reliability	Estimated Reliability
Org	5.05	5.08	5.60	1.0	0.000
ISM1	13.60	13.58	13.60	1.0	0.026
ISM3	9.50	9.50	10.00	1.0	0.000

Notes:
Org is the original network
ISM1 is the network after Investment Strategy Model 1
ISM3 is the network after Investment Strategy Model 3

Which investment strategy is better, and why?

As indicated in Table 11, the better investment strategy is to improve arc capacity. The reason for this is because Network 3 already had a path that had a reliability of 1 (the path between Nodes s, 1, 4, and t), thereby making it 100 percent reliable. Notice the expected reliability did not change for the increased reliability strategy. The reason for this is that the 100 percent reliable path already existed, and any further improvements in reliability increased the throughput, but not the expected reliability. The entire budget was spent for both strategies.

Table 12. Reliability Improvements For Arcs In Network 3

Start Node	Terminate Node	Initial Reliability	Reliability Increase	Final Reliability
s	2	0.1	0.9	1.0
2	4	0.5	0.5	1.0
4	3	0.0	0.5	0.5
3	t	0.9	0.1	1.0

Table 13. Capacity Improvements For Arcs In Network 3

Start Node	Terminate Node	Initial Capacity	Capacity Increase	Final Capacity
s	1	5	9	14
1	3	2	2	4
1	4	4	6	10
4	t	7	3	10

Given the budget available, which arcs are the best candidates for reliability improvements?

Table 12 shows the arcs that had their reliability improved. The reader should note that the potential arc from Node 4 to Node 3 was added. The potential arc was not added until the existing arcs had reached a reliability of 1. The reason for this is because the capacity of the potential arc is only 5 units, and increasing it's reliability has a small impact on the lower bound. By comparing the reliability increases in Table 12 to the diagram of the network in Figure 7 one can see that all the existing arcs reached a reliability of 1. Another important point is that the lower bound does not equal the upper bound in Table 11 even though the expected reliability is equal to 1. This occurs because there is flow along paths other than the 100 percent reliable path. Therefore, it is possible to increase the expected throughput as well as the lower and upper bounds by pushing flow through the other paths. The only time the lower bound will equal the upper bound is when all the arcs on the paths containing flow have a reliability of 1.

Given the budget available, which arcs are the best candidates for capacity improvements?

The arcs that were selected for capacity increases are listed in Table 13. Notice that the arcs along the 100 percent reliable path (Nodes s, 1, 4, and t) were increased

until their capacities equaled 10. Again this is because the flow through the path is limited by the arc with the smallest capacity. Because of this, the optimum solution for improving capacity must make sure that the capacities are the same along that path. Only one arc off the 100 percent reliable path had its capacity improved (the arc from Node 4 to Node t). A close analysis might have the reader wondering why the 3 units of capacity added to the arc from Node 4 to Node t was not just spread across the 3 arcs on the 100 percent reliable path The reason for this is that the objective function of Investment Strategy Model 1 maximizes the lower bound, and if the 3 units were given to the arcs on the 100 percent reliable path then the lower bound would have been 11 instead of 13.6.

What happens to the investment strategies when the cost of reliability is varied?

Network 3 was not sensitive to changes in reliability cost. By setting the budget at \$100 and capacity cost at \$5 in Investment Strategy Model 7, it was found that the price of reliability could vary from \$1 to infinity and have little effect on the lower bound. When the cost of reliability was set at \$1, the lower bound calculated by Investment Strategy Model 7 was 13.98. As the cost of reliability was increased, the lower bound decreased. The lower bound bottomed out at 13.6 when the cost of reliability reached \$6. Any cost value above \$6 resulted in the same lower bound of 13.6. Network 3 was insensitive to changes in reliability because it already had a 100 percent reliable path. In addition, the average path reliability of Network 3 was 0.65; this was much larger than any of the other networks.

8.4 Analysis of Network 4

Network 4 contains both potential arcs and stochastic nodes (see Figure 8). The stochastic nodes were converted into stochastic arcs before the models were run. The budget available for investment in Network 4 was \$1000 and the cost of increasing capacity by 1 or reliability by .1 was \$50. Network 4 can have up

Table 14. Original and Final Parameters of Network 4

Network	Lower Bound	Expected Throughput	Upper Bound	Expected Reliability	Estimated Reliability
Org	128.64	145.46	360.00	0.4353	0
ISM1	131.64	148.48	365.00	0.4353	0
ISM3	455.09	466.65	493.40	1.0000	0

Notes:
Org is the original network
ISM1 is the network after Investment Strategy Model 1
ISM3 is the network after Investment Strategy Model 3

to seven paths depending on the number of potential arcs added to the network. Only three paths exist in the original network. Once more, the lower bound was a better estimator of expected throughput than the upper bound, and the estimated reliability was zero because all paths contained flow. The results of the models for Network 4 are in Tables 14, 15, 16, 17, and 18, and the reports are in Appendix A.4.

Which investment strategy is better, and why?

The data in Table 14 shows that increasing reliability is the better investment strategy. There was only slight improvement to the final parameters when the capacity was increased while the final parameters of the increased reliability strategy was over threefold. In addition to increasing expected throughput from 145.46 to 466.65, the expected reliability increased from .4353 to 1.0 as a result of increasing reliability. The entire budget was expended in both investment strategies.

Given the budget available, which arcs are the best candidates for reliability improvements?

There are three arcs which had their reliability increased (see Table 15). Reliability was increased until the budget was exhausted. If the reader looks closely at Table 15, then he/she will see that the reliability increases are not in exact in-

crements of 0.1. This occurred because *GINO* will not solve for integer solutions in Investment Strategy Model 3. If an integer solution is desired, then Investment Strategy Model 3 can be used in conjunction with the Branch and Bound algorithm. Appendix E contains the integer solution to Network 4. However, since the experiment was designed to compare Investment Strategy Models 1 and 3, then the continuous solution shown in Table 15 will be used.

Given the budget available, which nodes are the best candidates for reliability improvements?

Table 16 contains the nodes which should have their reliabilities increased. The sum of the reliability increases in Tables 15 and 16 equals 2, which equates to total of 20 for the X_i variables. Since the cost of increasing reliability by .1 is \$50 and 20 units were purchased, the entire budget of \$1000 was spent. If additional funds were available then the lower and upper bounds, expected reliability, and expected throughput of the network could be improved even further by increasing reliability.

Given the budget available, which arcs are the best candidates for capacity improvements?

As shown in Table 17, only one arc, from Node 4 to Node 6, was selected for a capacity increase. This is because the most reliable path is along Nodes 1, 4, and 6. The only other arc on this path is from Nodes 1 to 4, and it had a capacity of 600

Table 15. Reliability Improvements For Arcs In Network 4

Start Node	Terminate Node	Initial Reliability	Reliability Increase	Final Reliability
4	6	0.600	0.400	1.000
3	5	0.300	0.517	0.817
5	6	0.700	0.283	0.983

Table 16. Reliability Improvements For Nodes In Network 4

Node Number	Initial Reliability	Reliability Increase	Final Reliability
3	0.700	0.117	0.817
4	0.500	0.500	1.000
5	0.800	0.183	0.983

(compared to 300 for the arc between Nodes 4 and 6). Therefore, the choke point along the path was the arc from Node 4 to 6.

Given the budget available, which nodes are the best candidates for capacity improvements?

Table 18 shows that only Node 4 was selected for a capacity increase. Notice that Node 4 lies on the same path as the arc that was selected for a capacity increase. This was to keep Node 4 from becoming the capacity choke point on the path. Another important point is that the capacity increases to the arc between Nodes 4 and 6, and to Node 4 itself are the same, i.e., 10 units. As a result, both the arc and node have a new capacity of 310.

What happens to the investment strategies when the cost of reliability is varied?

By leaving the cost of capacity fixed at $50 and the budget constant at $1000

Table 17. Capacity Improvements For Arcs In Network 4

Start Node	Terminate Node	Initial Capacity	Capacity Increase	Final Capacity
4	6	300	10	310

Table 18. Capacity Improvements For Nodes In Network 4

Node Number	Initial Capacity	Capacity Increase	Final Capacity
4	300	10	310

in Investment Strategy Model 7, it was discovered that increasing reliability was the better investment strategy until the cost of reliability reached \$6010. When the cost of reliability exceeded \$6010, then increasing capacity became the better strategy. Therefore, as long as the price of reliability is \$6010 dollars or less then the user should invest in reliability. Again just like in Networks 1 and 2, the path reliability of Network 4 was low, i.e. 0.1728, and this made increasing in reliability the better strategy.

8.5 Analysis of Network A

As stated in Chapter VI, Network A is more "realistic" than the four small networks. First of all, Network A has a total of 63 paths, which is by far more than any of the previous networks. Because of this high number of paths, it is possible that some paths may not be used at all. In addition, the topology of Network A is different from the others. As shown in Figure 9, there are two nodes which are very critical, Nodes 14 and 15. Since all of the flow must go through Node 14, it's reliability must be extremely high. The reliability of Node 15 is not as important as that of Node 14, but a large majority of the flow must go through Node 15. The budget for Network A was \$100,000 and the cost of increasing capacity by 1 or reliability by .1 was \$100. Tables 19, 20, 21, and 22 contain the results for Network A, and the actual reports are in Appendix A.9.

Which investment strategy is better, and why?

Table 19. Original and Final Parameters of Network A

Network	Lower Bound	Expected Throughput	Upper Bound	Expected Reliability	Estimated Reliability
Org	167	623	5760	0.1519	0.3352
ISM1	238	698	5760	0.1519	0.3501
ISM3	4266	9600	9600	1.0	0.0

Notes:
Org is the original network
ISM1 is the network after Investment Strategy Model 1
ISM3 is the network after Investment Strategy Model 3

As shown in Table 19, the better strategy is to increase reliability. The expected throughput for Investment Strategy Model 3 is 9600, while Investment Strategy Model 1 has only 698. In addition, the lower bound, upper bound, expected reliability, and estimated reliability are all higher for Investment Strategy Model 3. An important point to emphasize is that the solution to Investment Strategy Model 3 is only a local optimal. This occurred because of the gradient search algorithm used by *GINO* (see Chapter VII for details). The reason that the optimal is only a local can be seen by looking at Tables 19, 20, and 21. Due to the large budget, every arc and node in Network A had it's reliability increased to 1 (see Tables 20 and 21). This created a deterministic network, and therefore the lower bound, upper bound, and expected throughput in Table 19 should be equal for Investment Strategy Model 3 (just as it was in Networks 1 and 2). But, the lower bc nd for Investment Strategy Model 3 is much lower than the expected throughput and upper bound (4266 compared to 9600). This occurred because Investment Strategy Model 3 tried to push flow down every path (see Appendix A.9 for actual report), and this was not the global optimal. Even though the solution to Investment Strategy Model 3 is not the global, it is still much better than the optimal solution to Investment Strategy Model 1. It is possible to force *GINO* into finding other solutions by using the "GUESS"

command. The "GUESS" command allows the user of *GINO* to provide a starting value for any variable in the model. *GINO* will calculate the gradient from a new starting point based on the guesses provided by the user. For example, by using the "GUESS" command to set the flow through each path to that of the lower bound (flows through paths 1, 4, 24, 25, 36, 45, 54, and 61 all equaled 1200 in the lower bound, see Appendix A.4.3) for the original network, it was possible to increase the lower bound of Investment Strategy Model 3 from 4266 to 9434. Since the value 9434 is closer to the global optimal of 9600 than the solution calculated originally, i.e., 4266, then it is better than the original solution to Investment Strategy Model 3. Therefore, if the solution to Investment Strategy Model 3 is not the global solution, then it may be possible to improve the solution by setting the flow through each path equal to that of the lower bound. The estimated reliability (calculated by Equation 18) for Investment Strategy Model 3 is equal to zero because the local optimal pushes flow down every path.

Investment Strategy Model 3 spent $14,600 leaving a surplus of $85,400 while Investment Strategy Model 1 spent the entire budget.

Given the budget available, which arcs are the best candidates for reliability improvements?

Every arc in Network A had it's reliability increased to 1 (see Table 20). This occurred because of the size of the budget, i.e., $100,000.

Given the budget available, which nodes are the best candidates for reliability improvements?

Just like the arcs, the reliability of every node was increased to 1 because of the size of the budget. Table 21 shows how much reliability was added to each node.

Given the budget available, which arcs are the best candidates for capacity improvements?

Table 20. Reliability Improvements For Arcs In Network A

Start Node	Terminate Node	Initial Reliability	Reliability Increase	Final Reliability
2	5	0.3	0.7	1.0
2	14	0.6	0.4	1.0
5	10	0.6	0.4	1.0
5	11	0.7	0.3	1.0
6	14	0.6	0.4	1.0
7	14	0.6	0.4	1.0
8	14	0.3	0.7	1.0
10	15	0.6	0.4	1.0
12	15	0.7	0.3	1.0
14	17	0.3	0.7	1.0
14	18	0.6	0.4	1.0
14	19	0.6	0.4	1.0
15	6	0.3	0.7	1.0
15	7	0.6	0.4	1.0
15	8	0.7	0.3	1.0
17	16	0.7	0.3	1.0
18	16	0.6	0.4	1.0
19	16	0.7	0.3	1.0

As seen in Table 22, only one arc had it's capacity increased. It was the arc between Nodes 1 and 14. The increase was 1000 units, which used up the entire budget of $100,000.

What happens to the investment strategies when the cost of reliability is varied?

By leaving the cost of capacity fixed at $100 and the budget constant at $100,000 in Investment Strategy Model 7, it was discovered that increasing reliability was the better investment strategy until the cost of reliability reached $110,000. When the cost of reliability exceeded $110,000 then increasing capacity became the better strategy. Therefore, as long as the price of reliability is $110,000 dollars or

Table 21. Reliability Improvements For Nodes In Network A

Node Number	Initial Reliability	Reliability Increase	Final Reliability
2	0.3	0.7	1.0
3	0.7	0.3	1.0
4	0.5	0.5	1.0
5	0.8	0.2	1.0
7	0.3	0.7	1.0
8	0.7	0.3	1.0
9	0.5	0.5	1.0
10	0.8	0.2	1.0
12	0.3	0.7	1.0
13	0.7	0.3	1.0
14	0.5	0.5	1.0
15	0.8	0.2	1.0
16	0.8	0.2	1.0
17	0.7	0.3	1.0
18	0.3	0.7	1.0

less then the user should invest in reliability. This value is so high because the the average path reliability is only 0.0174. Since Network A is much larger than any of the others, it is not surprising that its path reliability is the lowest.

Table 22. Capacity Improvements For Arcs In Network A

Start Node	Terminate Node	Initial Capacity	Capacity Increase	Final Capacity
1	14	1200	1000	2200

8.6 *Lower Bound Versus Budget*

The relationship of the lower bound versus the budget available for investment is important in the analysis of the optimum investment strategy. It is possible to write the models formulated in Chapter III as Multiobjective Optimization Models by eliminating the budget constraint and replacing it with a second objective function. For example, Investment Strategy Model 7 can be reformulated as a Multiobjective Optimization Model as follows:

$$\text{Max } \sum_{j=1}^{q} R_j f_j \qquad \text{where } R_j = \prod_i (P_i + .1 \cdot X_i)$$

$$\text{Min } \sum_{i=1}^{n} (\alpha_{ci} d_i + \alpha_{ri} X_i)$$

s.t.

$$P_i + .1 \cdot X_i \leq 1 \qquad i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{q} a_{ij} f_j \leq u_i + d_i$$

$$X_i, P_i, f_j, \alpha_{ci}, \alpha_{ri} \geq 0 \text{ and } a_{ij} = 0, 1$$

The first objective is to maximize the lower bound, while the second is to minimize the amount of budget spent.

Obviously the feasible region for this multiobjective problem is more complex than the original single objective. De Neufville (9) describes a method, called the Constraint Method, to find the *noninferior* solutions to a multiobjective problem. A *noninferior* solution Y^* for a given set of variables, X, is one such that no other feasible solution is better than Y^* for the same X. The Constraint Method requires that one of the objective functions be changed into a constraint. By varying the right hand side of the constraint, it is possible to trace out the *noninferior* solutions (9:173-181). For a detailed description of multiobjective optimization, consult (9).

Figures 11, 12, 13, 14 and 15 show the relationship of the lower bound versus the budget amount for each of the sample networks. The graphs were generated by taking Investment Strategy Model 7, which already had the budget formulated as a constraint instead of an objective function, and varying the value of the budget. The

budget was first set to zero and the lower bound was calculated using *GINO*. The budget was then increased in discrete amounts until the reliability of the network equaled one. The cost of increasing capacity by 1 and reliability by .1 were the same. The lines in the graphs are the *noninferior* solutions. In addition, the slopes of the lines indicate how quickly the lower bound increases as the amount of the budget increases.

Figure 11 shows the graph of Network 1. The budget was set at zero and increased by \$10 increments until it reached \$100. Increasing reliability is the better investment strategy until the budget reaches \$50. When the budget reaches \$50, the overall network's reliability is one. As seen by the graph, the slope of the line is much greater for reliability improvements than for capacity improvements. This confirms the earlier finding that increasing reliability is the better strategy for Network 1. Once the network reaches a reliability of one then the lower bound shows only slight improvement for capacity increases.

The graph for Network 2 is contained in Figure 12. First, the budget was set to zero and then increased in \$100 increments until it reached \$2000. Increasing reliability is the better strategy until the budget reaches \$900, and then increasing capacity becomes the better strategy. Again it is possible to determine the better strategy by looking at the slope of line. The slope is much greater for reliability increases than capacity increases.

Figure 13 shows the graph of Network 3. This graph was generated by first setting the budget to zero and increasing it by \$10 increments until the budget reached \$200. Increasing capacity is the better strategy at every \$10 increment except \$60. This occurs because Network 3 is already highly reliable. Again this graph confirms the earlier finding that increasing capacity is the better strategy for Network 3.

The graph for Network 4 is contained in Figure 14. The budget was increased in \$100 increments until the it reached \$2500. Increasing reliability is the better

strategy until the budget reaches $1800. This is the point where the network's reliability equals one. The line shows some perturbations at $500 and $1200 where the slope changes. These occurs because the reliability of specific paths is equal to one.

Figure 15 shows the graph for Network A. The budget was increased in $1000 increments until it reached $20,000. Increasing reliability is the better strategy until the budget reaches $8,000. Between $9,000 and $14,000 only 24 percent of the budget is spent on reliability, and the rest is spent on capacity. Another large purchase is made in reliability between $14,000 and $15,000. The reason for this behavior is because the capacities of the bottleneck arcs are being increased to a point where increasing reliability is once again the better strategy. At $15,000 the network's reliability is equal to one. The better strategy between $15,000 and $20,000 is to increase capacity.

Figure 16 contains all the graphs on a single page. This figure shows that the general shape of the curves for all the networks is the same. The slopes of the lines show that the lower bound will increase rapidly until the overall network reliability nears one, and will then increase slowly at a rate equal to the cost of capacity.

8.7 Summary

Investment Strategy Models 1 and 3 were run against Networks 1, 2, 3, 4, and A. Investment Strategy Model 1 maximizes the lower bound of expected maximum flow by increasing capacity, while Investment Strategy Model 3 increases reliability to maximize the lower bound. After each investment strategy was calculated for the networks, then an analysis was done to see which was better.

The better investment strategy depends on the topology of the network being analyzed. On four of the Networks (1, 2, 4, and A), it was obvious that increasing reliability was the better strategy. However, the better strategy for Network 3 was to increase capacity. The reason Network 3 had its capacity increased in lieu of

reliability was because it already had a path that was 100 percent reliable. Since the path was operational all the time, then there was no need to improve reliability. The other four networks did not have a 100 percent reliable path, and as a result the reliability increase was the better investment. In addition, as the size of the networks increased then it became obvious that increasing reliability was the better investment. This occurred because the average path reliability decreased as the networks got larger.

In the cases of Networks 1, 2, and A, there was a surplus left in the reliability budget. This occurred because all the arcs and nodes reached a reliability of 1 before the budget was expended, and there was no need to spend any more money. On the other hand, the Investment Strategy Model 1 always spent the entire capacity budget since there was no limit on arc capacity.

The lower bound was a much better estimator of expected throughput than the upper bound. This implies that the expected throughput will always be on the pessimistic side. This is an important fact and cannot be stressed enough. Since the lower bound can be computed much quicker than the amount of time required to run the Monte Carlo simulation program for calculating expected throughput, it seems that computing the lower bound is an efficient alternative to simulation.

Another fact that became apparent on the sample networks was that the estimated reliability equation (Equation 18) does not work on small networks. The estimator consistently underestimated the expected reliability. This occurred because the small networks had only a few paths and they always contained flow. In a larger network with many paths, it is possible to have paths which do not have any flow, thereby making Equation 18 a better estimator of network reliability.

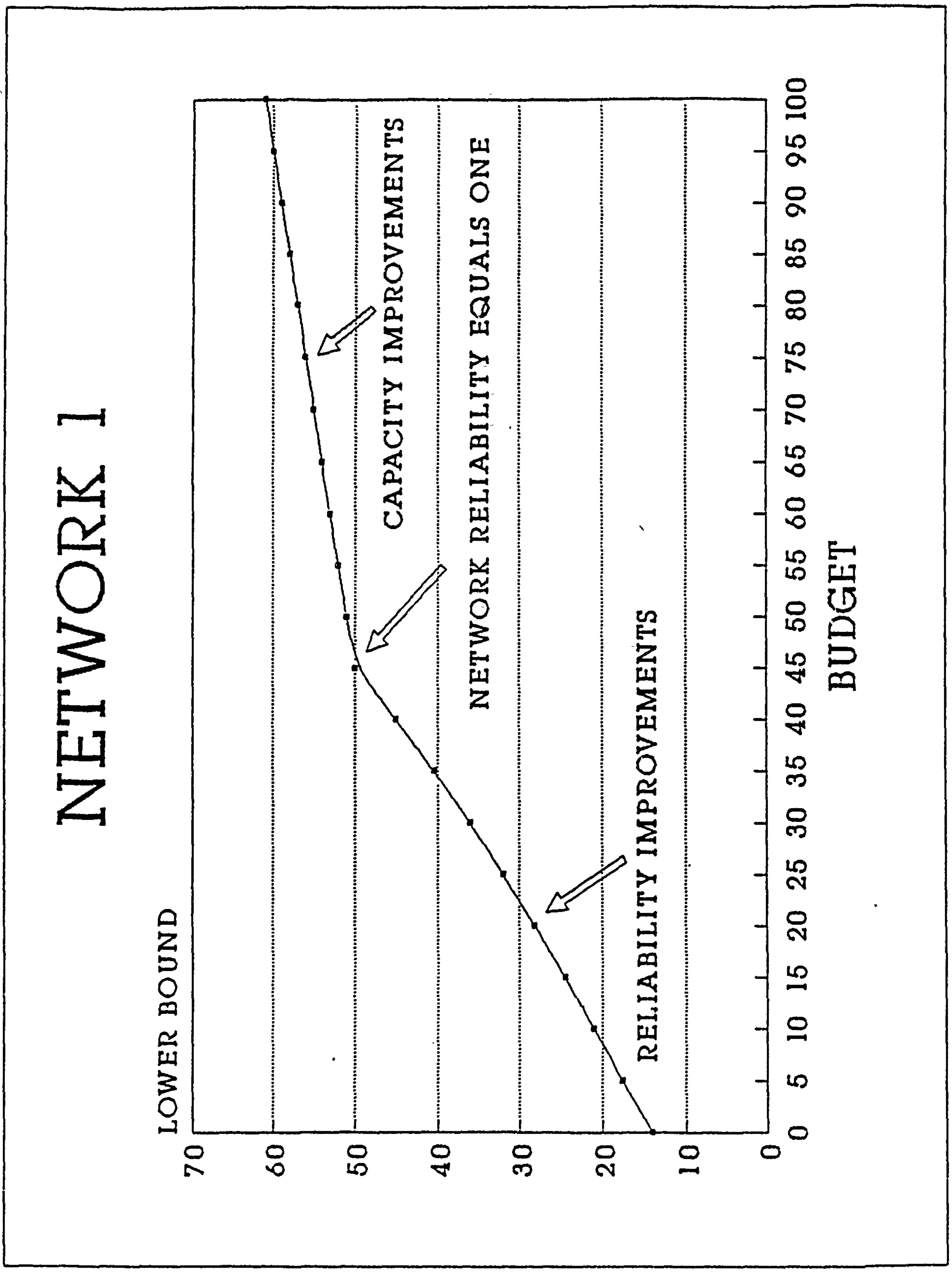


Figure 11. Plot of Network 1

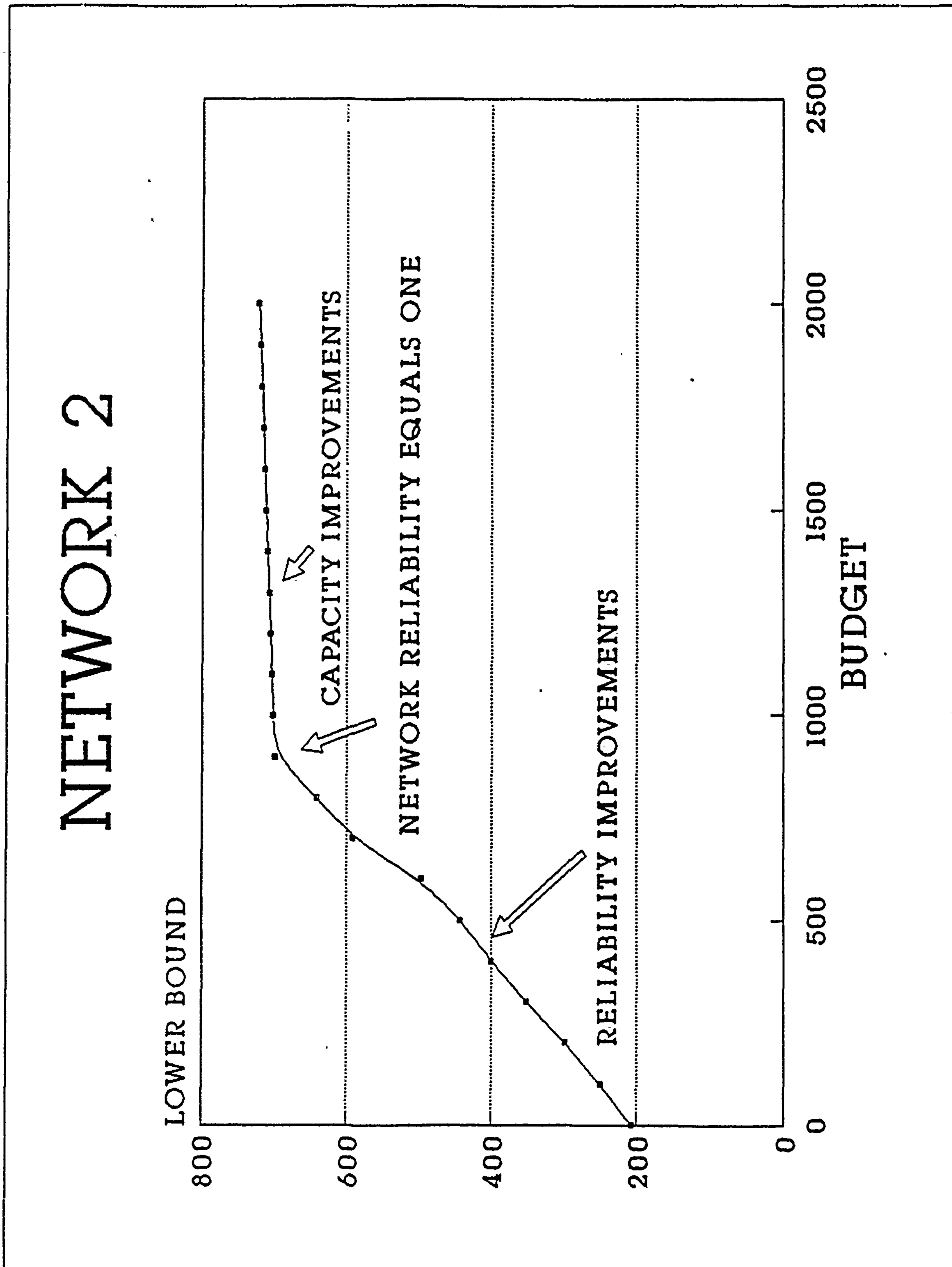


Figure 12. Plot of Network 2

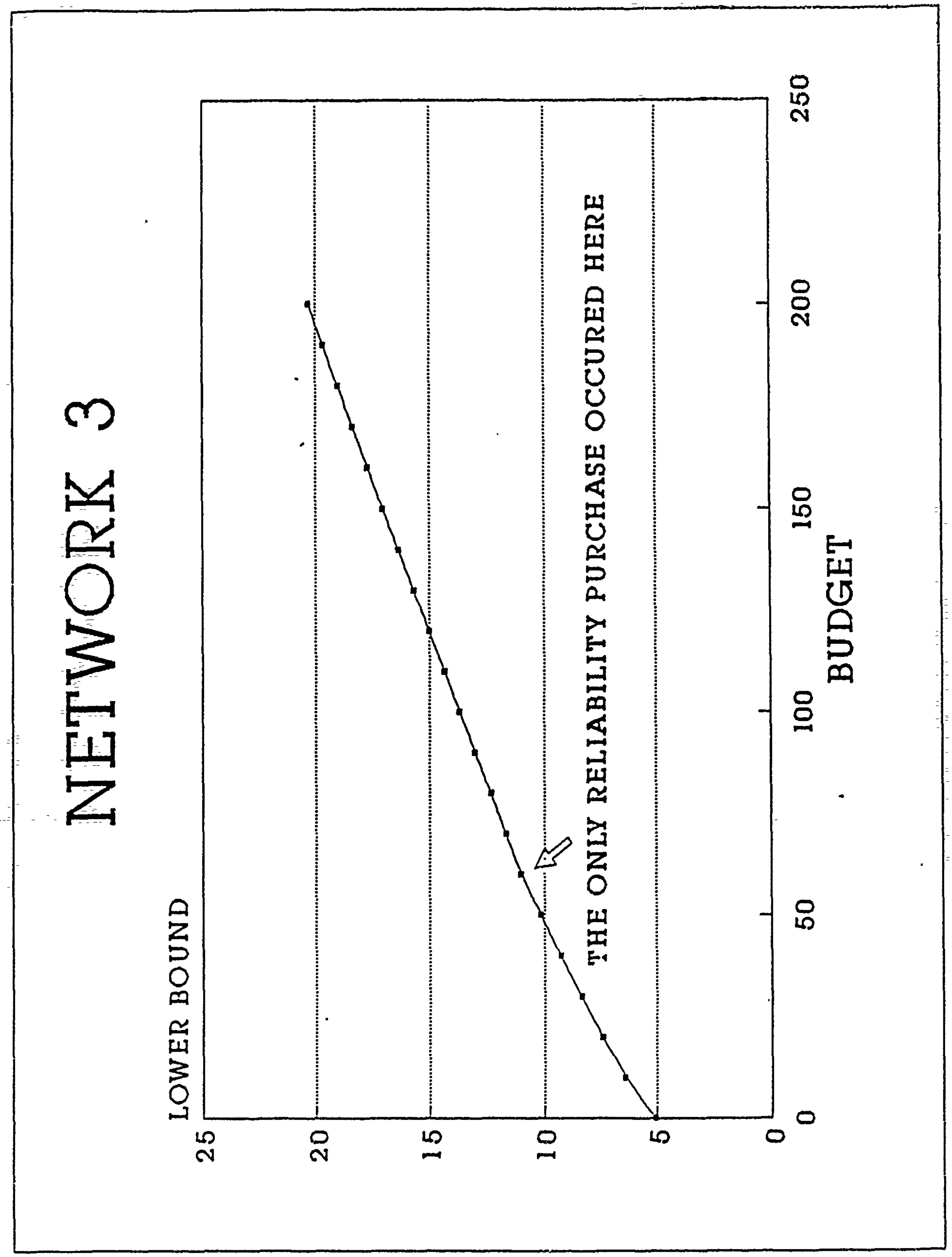


Figure 13. Plot of Network 3

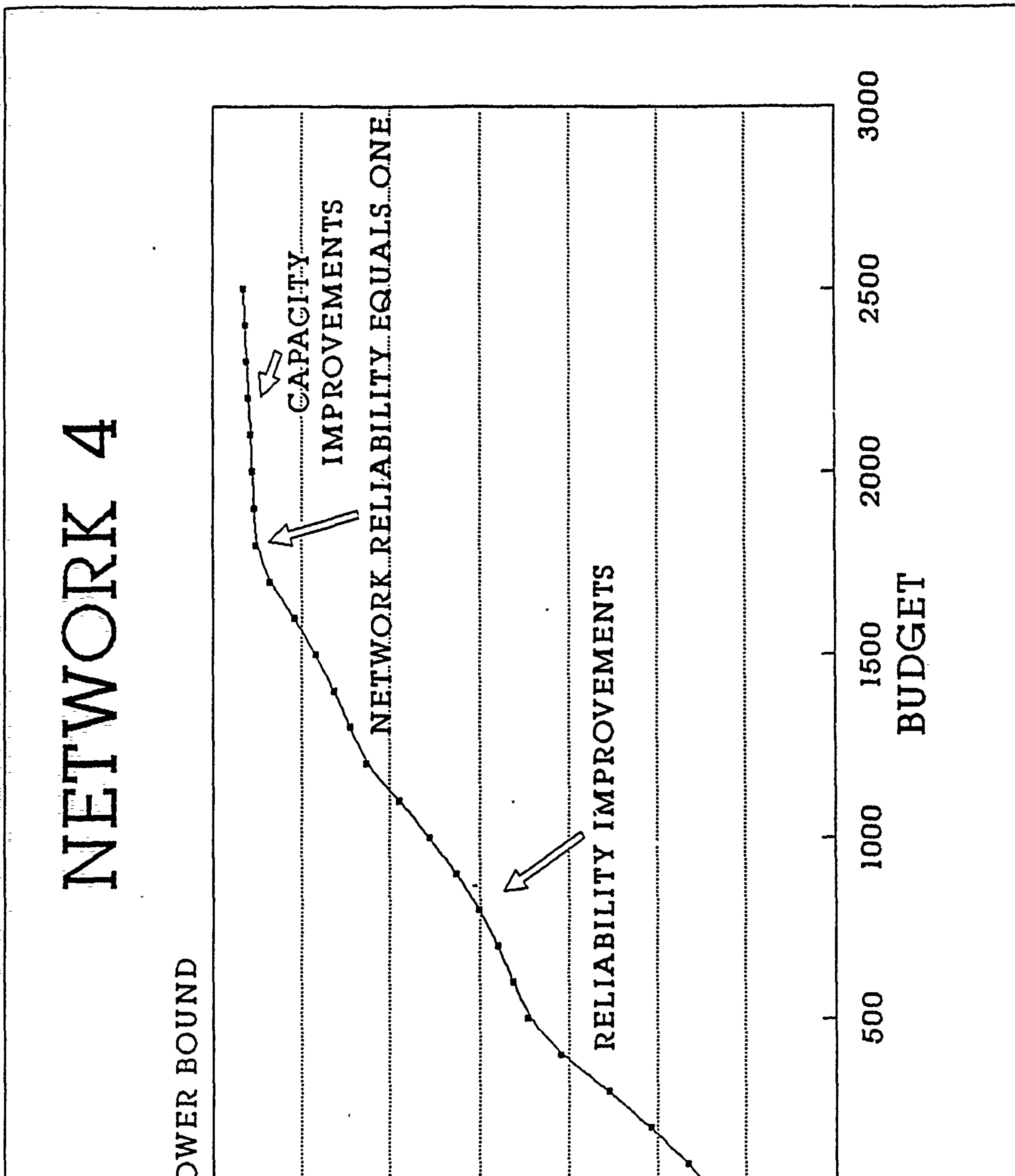


Figure 14. Plot of Network 4

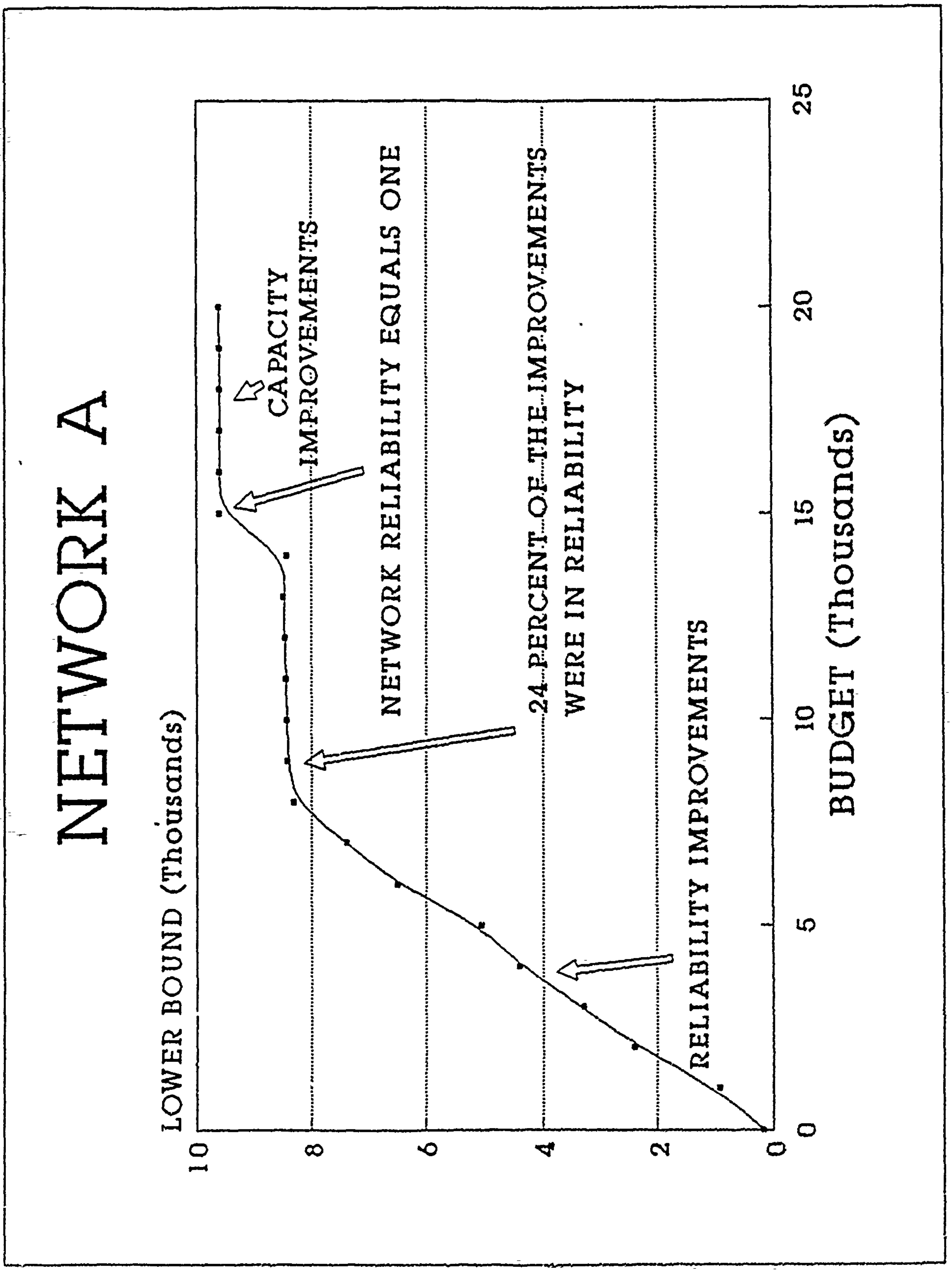


Figure 15. Plot of Network A

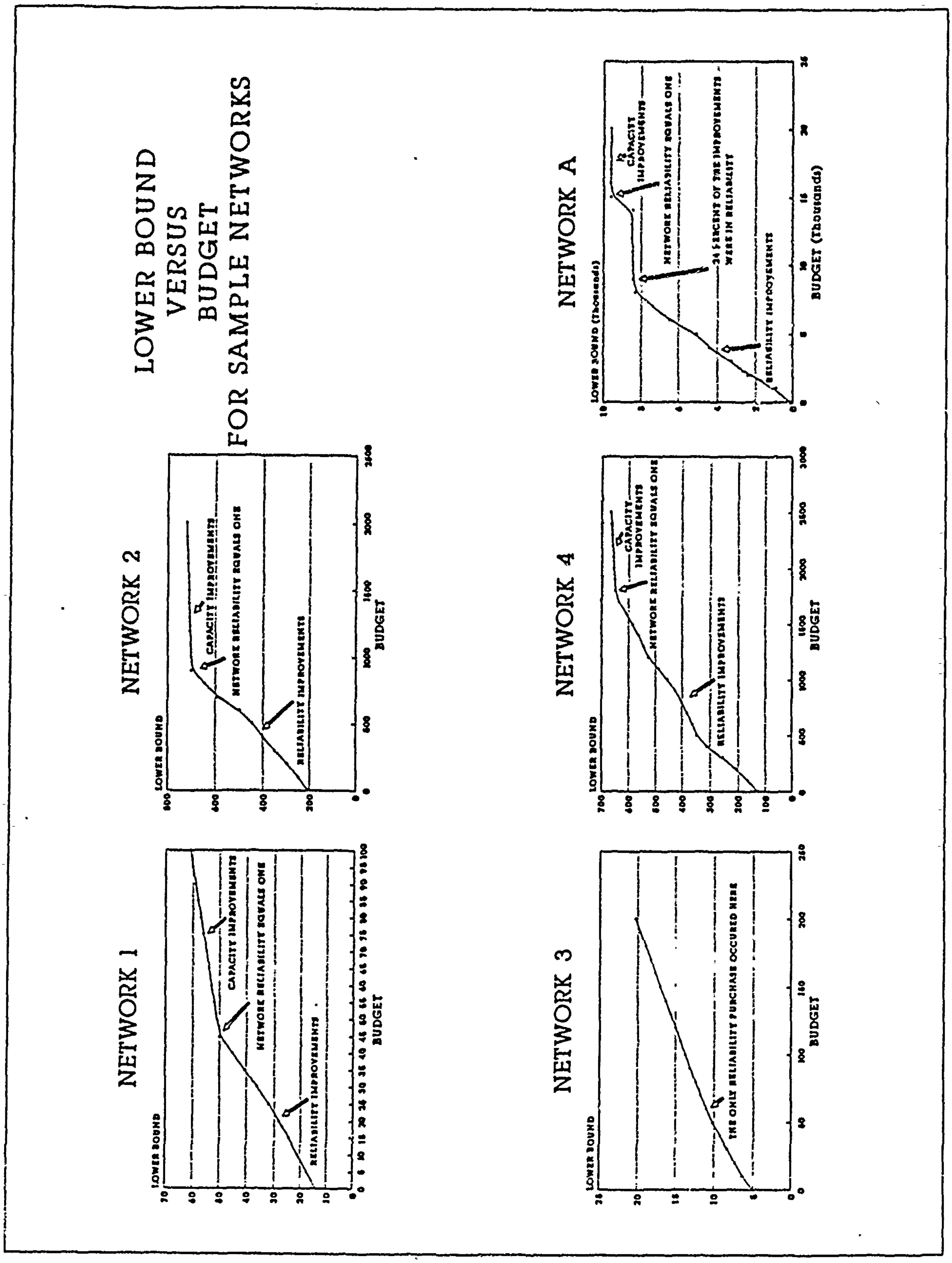


Figure 16. Comparison of Network Plots

IX. Solving Models Using Network Packages

The models formulated in Chapters II and III required the use of either a linear or nonlinear programming package, depending on the model being solved. While these packages are straight forward and easy to use, they may not be the most efficient. Network packages are an alternative to linear programming packages. They are designed specifically for network type problems. Many types of problems can be formulated as networks, and therefore use the more efficient algorithms that are embedded in the network packages.

Some of the models in Chapters II and III are formulated as linear programs using the arc-path incidence matrix. Network packages recognize the fact that a network structure can be embedded in a linear program. This type of formulation is called a network with side constraints (20:359). The following sections describe how, with a small amount of modification, the Lower and Upper Bound Models, Investment Strategy Model 1 and Investment Strategy Model 4 can be solved using network packages.

9.1 Decision Variables

The following decision variables will be used to formulate the models:

R_k	=	the reliability of path k
f_k	=	the flow through path k
f_{ij}	=	the flow through the arc between nodes i and j
X_{ij}	=	the number of .1 increments of reliability added to the arc between nodes i and j
p_{ij}	=	the initial reliability of the arc between nodes i and j

d_{ij} = the increase in capacity of the arc between nodes i and j

u_{ij} = the initial capacity of the arc between nodes i and j

a_{ij} = equals 1 if the arc between nodes i and j lies on path k; 0 otherwise

α_{ij} = the cost of increasing capacity by 1 or reliability by .1 for the arc between nodes i and j

β = the total amount of the budget

9.2 Solving Lower and Upper Bound Models

The Lower and Upper Bound Models formulated in Section 2.5 used the arc-path incidence matrix, and were solved using a linear programming package called LP/MIP-83. It is possible to reformulate the Lower and Upper Bound models by adding side constraints to the Maximum Flow Model. The Maximum Flow Model calculates the maximum flow through a network that is not stochastic, i.e., the reliabilities are equal to one. A formulation of the Maximum Flow Model using the node-arc incidence matrix follows:

$$\text{Max } V$$

s.t.

$$\begin{aligned}
&\textstyle\sum_{j\in S^+(i)} f_{ij} - \sum_{j\in S^-(i)} f_{ij} = \text{-}V && \text{source node}\\
&\textstyle\sum_{j\in S^+(i)} f_{ij} - \sum_{j\in S^-(i)} f_{ij} = 0 && \text{intermediate nodes}\\
&\textstyle\sum_{j\in S^+(i)} f_{ij} - \sum_{j\in S^-(i)} f_{ij} = V && \text{sink node}\\
&0 \le f_{ij} \le u_{ij}
\end{aligned}$$

Notation: $S^+(i)$ are the arcs terminating at node i and $S^-(i)$ are the arcs initiated at node i

where f_{ij} and u_{ij} are the flow and flow capacity between nodes i and j respectively (23:31). The first three constraints make up the node-arc incidence matrix and the last constraint is called a side constraint. It restricts the flow on each arc to be equal to or less than the capacity of the arc. There are many commercial packages, both linear programming and network, available to solve the Maximum Flow Model.

9.2.1 Lower Bound Solution Using Netflow. The lower bound can be calculated using a network package that supports side constraints. By adding additional side constraints to the Maximum Flow Model, it is possible to solve the Lower Bound Model more efficiently using a network package. This researcher chose to use a commercial package called *SAS*. *SAS* is a software program that consists of numerous modules, called procedures, to solve analytical models. One of the procedures is called *Netflow*. This procedure will solve both minimum cost and maximum flow problems that are formulated as networks with side constraints (20:358). A formulation of the Lower Bound Model using the Maximum Flow Model augmented with side constraints follows:

$$\text{Max } V$$

s.t.

$$\begin{array}{llll}
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = -V & \text{source node} \\
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = 0 & \text{intermediate nodes} \\
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = V & \text{sink node} \\
\sum_{k=1}^{q} a_{ij} f_k & \leq u_{ij} & & \\
\sum_{k=1}^{q} R_k f_k & -\sum_{i\in S^+(t)} f_{it} & = 0 &
\end{array}$$

Notation: $S^+(t)$ are the arcs terminating at node t

and q is the number of paths in network

where a_{ij} equals 1 if the arc from node i to node j lies on path k, and 0 otherwise. The variables f_k and u_{ij} represent the flow on path k and the capacity of the bottleneck arc between nodes i and j that lies on path k, respectively. Variable f_{it} is the flow going into node t from node i. The last constraint limits the flow through the network to the that of the lower bound. Upon careful observation, the reader will notice that the last side constraint consists of the objective function from the Lower Bound Model formulated in Section 2.5.1, minus the total flow coming into the sink node t. Therefore, this constraint limits the flow into t to be no more than the lower bound. By adding this side constraint, it is possible to solve for the lower bound

using a network package instead of a linear programming package. This model is called Lower Bound Formulation 2.

The model was run on Networks 2, 3, and 4 as they are shown in Figures 6, 7, and 8. The model could be run on Network A, but the manual preparation for input into *Netflow* would be very cumbersome and time consuming. (A recommendation that *Formula* Version 2.0 be enhanced to formulate the models in this chapter is made in Chapter X). The results from *Netflow*, as well as the lower bounds calculated by the arc-path formulation using a linear program package LP/MIP-83 are presented in Table 23 (the LP/MIP-83 values were taken from Tables 4, 7, and 10). As expected, the lower bounds calculated by LP/MIP-83 and *Netflow* are the same (the reports from *Netflow* are contained in Appendix A.5). Since *Netflow* is designed specifically for network problems, it calculated the results faster.

Table 23. Lower Bound Comparison

Network	Value from LP/MIP-83	Value from *Netflow*
2	206.0	206.0
3	5.05	5.05
4	128.64	128.64

9.2.2 Solving Upper Bound Using Either Netflow or Maxflo. The upper bound is a special case of the maximum flow problem where the expected capacity of an arc is equal to the reliability of the arc times it's capacity ($p_{ij} \cdot u_{ij}$). Making this change to the Maximum Flow Model results in the following node-arc formulation of the upper bound:

$$
\begin{array}{lllll}
\text{Max } V & & & & \\
\text{s.t.} & & & & \\
\sum_{j \in S^+(i)} f_{ij} & - \sum_{j \in S^-(i)} f_{ij} & = -V & \text{source node} \\
\sum_{j \in S^+(i)} f_{ij} & - \sum_{j \in S^-(i)} f_{ij} & = 0 & \text{intermediate nodes} \\
\sum_{j \in S^+(i)} f_{ij} & - \sum_{j \in S^-(i)} f_{ij} & = V & \text{sink node} \\
0 \leq f_{ij} \leq p_{ij} \cdot u_{ij} & & &
\end{array}
$$

This model is called Upper Bound Formulation 2. Note, constraint number four limits the flow through the arc from nodes i to j to be no more than the expected capacity. This model can be solved using either a standard network package or by a network package that supports side constraints. To calculate the upper bound using a standard networking package one merely inputs the expected capacity value, $(p_{ij} \cdot u_{ij})$, when prompted for the capacity of the arc from node i to node j.

The upper bound can also be solved using a Min Cost model. A Min Cost model for solving the upper bound is formulated by adding an arc from the sink node, t, to the source node, s. The cost of sending one unit of flow down the arc from t to s is -1, and the cost of sending one unit of flow on the remaining arcs is zero. Again, the maximum flow through an arc is the expected capacity. Below is the upper bound formulated as a Min Cost model:

$$
\begin{array}{llllll}
\text{Min } -X_{ts} & & & & \\
\text{s.t.} & & & & \\
\sum_{j \in S^+(i)} f_{ij} & - \sum_{j \in S^-(i)} f_{ij} & +X_{ts} & = 0 & \text{source node} \\
\sum_{j \in S^+(i)} f_{ij} & - \sum_{j \in S^-(i)} f_{ij} & & = 0 & \text{intermediate nodes} \\
\sum_{j \in S^+(i)} f_{ij} & - \sum_{j \in S^-(i)} f_{ij} & -X_{ts} & = 0 & \text{sink node} \\
0 \leq f_{ij} \leq p_{ij} \cdot u_{ij} & & & &
\end{array}
$$

This model is called Upper Bound Formulation 3, and can be solved using either a linear programming package or a network package.

Table 24. Upper Bound Comparison

Network	Value from LP/MIP-83	Value from *Netflow*	Value from *Microsolve*
2	380	380	380
3	5.6	5.6	5.6
4	360	360	360

The upper bound was found using both Upper Bound Formulations 2 and 3. First, *Netflow* was used to solve Upper Bound Formulation 2, and *Microsolve* was used to calculate the upper bound for Upper Bound Formulation 3. *Microsolve* is a package that runs on an IBM-compatible microcomputer. It supports both linear programming and network models (11:39-55). Table 24 contains the upper bounds as they were calculated by LP/MIP-83, *Netflow*, and *Microsolve*. As expected, the values are the same for all three models. The values for LP/MIP-83 came from Tables 4 7, and 10. The actual reports from *Microsolve* are in Appendix A.6. Again, the network packages were more efficient than the linear programming package since they are designed for network problems.

9.3 Solving Investment Strategy Model 1 Using Netflow

Investment Strategy Model 1 was formulated in Section 3.3 using the arc-path incidence matrix. However, since Investment Strategy Model 1 maximizes the lower bound by increasing arc capacity it is possible to reformulate the model using the Maximum Flow Model and adding side constraints. The formulation is accomplished by modifying the Lower Bound Formulation 2 model presented in Section 9.1.1 as seen below:

Max V

s.t.

$$\begin{array}{llll}
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = -V & \text{source node} \\
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = 0 & \text{intermediate nodes} \\
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = V & \text{sink node} \\
\sum_{k=1}^{q} a_{ij} f_k & \leq u_{ij} + d_{ij} & & \\
\sum_{k=1}^{q} R_k f_k & -\sum_{i\in S^+(t)} f_{it} & = 0 & \\
\sum_{k=1}^{q} \alpha_{ij} d_{ij} & \leq \beta & &
\end{array}$$

Notation: $S^+(t)$ are the arcs terminating at node t

and q is the number of paths in the network

The variable d_{ij} is the amount of capacity added to the arc from Node i to Node j. In addition, α_{ij} is the cost of one additional unit of capacity from Node i to Node j, and β is the amount of the budget available for investing. This model is called Investment Strategy Model 1 Formulation 2, and can be solved using *Netflow*. Table 25 shows the optimum lower bounds that result when both formulations of Investment Strategy Model 1 were run against Networks 2, 3, and 4. The LP/MIP-83 values came from Tables 4, 7, and 10, and were calculated using the arc-path formulation of Investment Strategy Model 1 presented in Section 3.3.

Table 25. Investment Strategy Model 1 Comparison

Network	Value from LP/MIP-83	Value from *Netflow*
2	210.2	210.2
3	13.6	13.6
4	131.64	131.64

Not only do the optimum values of the lower bound match as pointed out in Table 25, but the investment strategies are also the same (the *Netflow* reports are

in Appendix A.7). For example, in Network 2 the *Netflow* report indicated that the capacity of the arcs between Nodes 3 and 5, and Nodes 5 and 6 should each be increased by 10 units (see Appendix A.7). This matches the values in Table 6, which were computed by LP/MIP-83.

9.4 Solving Investment Strategy Model 4 Using Netflow

Investment Strategy Model 4 is a arc-path formulation for maximizing the upper bound by improving arc reliability (Investment Strategy Model 4 is formulated in Section 3.6). Instead of using the arc-path formulation and solving the model using a linear programming package, it is possible to reformulate the model using the Maximum Flow Model and side constraints. The model would attempt to maximize the upper bound by increasing the reliability of arcs. The amount of increase is subject to a budget constraint. The model with side constraints looks like this:

$$\text{Max } V$$

s.t.

$$\begin{array}{llll}
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = \text{-}V & \text{source node} \\
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = 0 & \text{intermediate nodes} \\
\sum_{j\in S^+(i)} f_{ij} & -\sum_{j\in S^-(i)} f_{ij} & = V & \text{sink node} \\
0 \leq f_{ij} \leq (p_{ij} + .1X_{ij}) \cdot u_{ij} & & & \\
p_{ij} + .1X_{ij} \leq 1 & & & \\
\sum \alpha_{ij} X_{ij} \leq \beta & & &
\end{array}$$

Where α_{ij} is the cost of increasing the reliability of the arc from Node i to Node j by .1, X_{ij} is a continuous variable that denotes the number of units of reliability to buy, and β is the amount of the investment budget. This model is called Investment Strategy Model 4 Formulation 2.

The first three constraints make up the node-arc incidence matrix, while the last three are the side constraints. *Netflow* can be used to solve the model. The solutions to the model were calculated for Networks 2, 3, and 4. The results are

presented in Table 26 (the actual reports are in Appendix A.8). As expected, the results match those from LP/MIP-83 (see Tables 4, 7, and 10 in Chapter VIII).

Table 26. Optimum Upper Bound Using *Netflow*

Network	Upper Bound
2	700
3	10
4	650

9.5 Comparing Investment Strategy Model 3 with Investment Strategy Model 4

The experiment described in Chapter VII generated the data so a comparison could be made between Investment Strategy Model 1 and Investment Strategy Model 3. The results and analysis of the experiment are presented in Chapter VIII. Both models maximized the lower bound of throughput. This section describes the results of an experiment that compares maximizing the lower bound to maximizing the upper bound when the reliability of arcs is increased.

The two models that were used are Investment Strategy Model 3 and Investment Strategy Model 4 Formulation 2. Since Investment Strategy Model 3 was used in the earlier experiment, the results from maximizing the lower bound were already available. Therefore, the only model left to run was Investment Strategy Model 4 Formulation 2. *Netflow* was used to solve this model on Networks 2, 3, and 4 as they appear in Figures 6, 7, and 8 respectively. The results for Investment Strategy Model 3 were taken from Tables 4 thru 14 in Chapter VIII, and the *Netflow* reports for Investment Strategy Model 4 Formulation 2 are contained in Appendix A.8.

Table 27 shows the lower and upper bounds, expected throughput, and expected reliability for Networks 2, 3, and 4 as a result of implementing the investment strategies calculated by the two models.

Table 27. Investment Strategy Model 3 Versus Investment Strategy Model 4

Investment Strategy Model 3				
Network	Lower Bound	Expected Throughput	Upper Bound	Expected Reliability
2	700.0	700.0	700.0	1.0000
3	9.5	9.5	10.0	1.0000
4	455.09	466.65	493.4	1.0000
Investment Strategy Model 4				
Network Network	Lower Bound	Expected Throughput	Upper Bound	Expected Reliability
2	620.00	620.680	700.0	1.0000
3	8.3732	8.3669	10.0	1.0000
4	401.36	420.332	650.0	1.0000

After examining the data in Table 27, one critical fact emerges. The expected throughput computed by the Monte Carlo simulation program *Maxflo* is higher when Investment Strategy Model 3 is used. This means that on the average, Investment Strategy Model 3 will result in a higher network throughput than Investment Strategy Model 4. Since Investment Strategy Model 3 maximizes the lower bound, this confirms the earlier analysis that the lower bound is a better estimator of throughput than the upper bound. The reason that the expected throughput is lower for Investment Strategy Model 4 on each network in Table 27 is that Investment Strategy Model 4 will give priority to increasing reliability on the bottleneck arc of a path, while Investment Strategy Model 3 increases overall path reliability. As a result, the overall path reliability may not be increased as much by Investment Strategy Model 4. In addition, Investment Strategy Model 4 may not spend the entire budget, even though there are arcs that could have their reliability increased.

9.5.1 Investment Strategies for Network 2. Table 28 shows the different investment strategies for the Network 2. The investment budget was $1000, and the

Table 28. Reliability Improvements For Arcs In Network 2

Investment Strategy Model 3			Investment Strategy Model 4		
Start Node	Terminate Node	Reliability Increase	Start Node	Terminate Node	Reliability Increase
2	4	0.5	2	4	0.5
4	6	0.6	4	6	0.4
3	5	0.3	3	5	0.3
5	6	0.4	5	6	0.4

cost of increasing reliability by .1 was $50. The only difference between the two investment strategies is the amount of reliability to add to the arc going from Node 4 to Node 6. Investment Strategy Model 3 increased the arc by 0.6, while Investment Strategy Model 4 increased it by 0.4. This is because Investment Strategy Model 4 gives priority to the bottleneck arcs. The bottleneck arc for the top path in Network 2 (Nodes s, 1, 2, 4, 6, and t), as shown in Figure 6, is between Nodes 2 and 4; it has a capacity of 400. When the reliability of this arc reaches 1, then the expected capacity is equal to 1 times 400, or just plain 400. Once the reliability of the bottleneck arc reaches 1, then the remaining arcs on the path will have their reliability increased until their expected capacity reaches that of the bottleneck arc. For example, the only other arc that can have it's reliability increased on the top path is between Nodes 4 and 6. Its reliability was increased from 0.4 to 0.8; this gives it an expected capacity of 400 also (0.8 times 500 = 400). Investment Strategy Model 3 increased the reliabilities until all arcs had a reliability of 1. This strategy cost a total of $900. On the other hand, Investment Strategy Model 4 only spent $800 since it increased the reliability of the arc from Node 4 to 6 by 0.4 instead of the full 0.6 available. The *Netflow* reports for Investment Strategy Model 4 are in Appendix A.8.

Table 29. Reliability Improvements For Arcs In Network 3

Investment Strategy Model 3			Investment Strategy Model 4		
Start Node	Terminate Node	Reliability Increase	Start Node	Terminate Node	Reliability Increase
s	2	0.9	s	2	0.733
2	4	0.5	2	4	0.5
4	3	0.5	4	3	0.32
3	t	0.1			

9.5.2 Investment Strategies for Network 3. Table 29 shows the investment strategies for Network 3. The budget was set at \$100, and the cost of increasing reliability by .1 was \$5. Again the reliability increases are different because of the priority that Investment Strategy Model 4 gives the bottleneck arcs (see reports in Appendix A.8 for details). The detailed discussion is omitted since the models used the same logic discussed in the analysis of Network 2.

9.5.3 Investment Strategies for Network 4. Tables 30 and 31 show the investment strategies for Network 4. The nodes in Network 4 are stochastic, and therefore their reliabilities can be increased. Again the logic of the investment strategies are the same as that of Network 2. The reports from *Netflow* are contained in Appendix A.8.

9.6 Summary

This chapter shows that the linear models presented in Chapters II and III, which were based on the arc-path incidence matrix, can be reformulated using the node-arc incidence matrix. By augmenting the Maximum Flow Model with a set of side constraints, it is possible to solve the linear models presented in Chapters II and III using a commercial network package. Network packages are software

Table 30. Reliability Improvements For Arcs In Network 4

Investment Strategy Model 3			Investment Strategy Model 4		
Start Node	Terminate Node	Reliability Increase	Start Node	Terminate Node	Reliability Increase
4	6	0.400	4	6	0.400
3	5	0.517	1	5	0.300
5	6	0.283	5	6	0.300

programs specifically designed to solve network models. They are more efficient than linear programming packages, and it is possible to analyze larger networks using the network packages. The following models were reformulated using the Maximum Flow Model with side constraints: Lower Bound, Upper Bound, Investment Strategy Model 1, and Investment Strategy Model 4. The network package *Netflow* was used to solve all the new formulations. Another network package called *Microsolve* was used to solve the upper bound as a min-cost problem.

In addition, a comparison was made between Investment Strategy Model 3 and Investment Strategy Model 4. Both of these models increase arc reliability, but their objective functions are different. Investment Strategy Model 3 optimizes the lower bound, while Investment Strategy Model 4 optimizes the upper bound. The data

Table 31. Reliability Improvements For Nodes In Network 4

Investment Strategy Model 3		Investment Strategy Model 4	
Node Number	Reliability Increase	Node Number	Reliability Increase
3	0.117	2	0.300
4	0.500	4	0.500
5	0.183		

resulting from implementing the separate investment strategies on Networks 2, 3, and 4 indicated that on the average the expected throughput would be higher when Investment Strategy Model 3 was used.

IV, the arc reliabilities of the networks are the driving force behind which strategy is better. If the network has a path that is 100 percent reliable then the better strategy is to improve capacity. However, if the network does not contain a 100 percent reliable path, which is probably the case in a "real world" network, then the better strategy was to improve reliability. Based on this small sample of network topologies and the assumptions stated in the earlier chapters, it appears that increasing reliability would be the better strategy to improve throughput in a stochastic communications network. In addition, as the networks increased in size, the average path reliability decreased. Because of this relationship, the better strategy for large networks is to improve reliability.

2. Of the two estimators for expected throughput, lower and upper bound, it is apparent that the lower bound represents a more realistic value of the throughput. Again, this conclusion is based on the analysis of the sample networks. Since the lower bound represents a pessimistic estimate of the throughput, it would appear that the expected throughput will generally be on the low side. For this reason, a conservative estimate should be used when analyzing the throughput of a stochastic communications network.

3. The equation to calculate network reliability (Equation 18 in Chapter VIII) is not a good estimator for small networks.

4. The data in Chapter IX indicates that on the average, the optimum investment strategy calculated by Investment Strategy Model 3, maximizing the lower bound, will result in a higher expected throughput than that of Investment Strategy Model 4, maximizing upper bound. Therefore, it is better to improve the lower bound rather than the upper bound.

5. Since *GINO* will only solve for a local optimal, it may not be possible to calculate a global optimal for Investment Strategy Model 3. This means that

the investment strategy for improving reliability may not be the best that is available to the investor.

6. It is possible to formulate the lower bound using either the arc-path or node-arc incidence matrices. However, since the arc-path formulation requires a linear programming package, which must continuously update the matrix inverse, to solve for the lower bound, it is less efficient than the node-arc formulation. Therefore, the node-arc formulations for the lower bound and Investment Strategy Model 1 are superior than the arc-path formulations of Yim (23).

10.2 Recommendations

The following recommendations are made for extending the research in this document:

1. An in-depth analysis should be performed on larger networks to see if the conclusions reached above still hold.

2. Instead of calculating the reliability of every path in the network, a study should be made in finding an algorithm for calculating the most probable states of stochastic communications networks.

3. Enhance Investment Strategy Models 1, 2, 3, 4, 5, 6, and 7 so that multicommodity flows can be analyzed. As stochastic communications networks start to send a wider variety of information, i.e., voice, computer data, fax, etc., it becomes more apparent that throughput will be affected.

4. Enhance Investment Strategy Models 3, 4, 5, 6, and 7 to pick the best locations for potential arcs instead of the user having to determine them. An algorithm using artificial intelligence may have to be developed.

5. Rewrite *Formula* Version 2.0 so that it will output models which can be input directly into *Netflow*.

6. Investigate the use of the Branch and Bound algorithm and Lagrangian Relaxation for solving Investment Strategy Models 2, 4, 6, and 7 to provide integer solutions.

10.3 Summary

Further research into the area of stochastic communications networks is needed as the networks become more complex. The throughput of stochastic communications networks will be an ongoing concern of DoD, and the study of such networks is crucial to the security of the United States.

Appendix A. *Reports for Sample Networks*

A.1 Network 1

A.1.1 Formula Input File

```
% test1.for

arc(s,1).
arc(1,2).
arc(2,3).
arc(3,t).

prob(s,1).
prob(1,0.4).
prob(2,0.7).
prob(3,1).
prob(t,1).

cap(1,100).
cap(2,50).
cap(3,*).

cost(_,5).

invest(_,0).

budget(100).

rcost(_,5).

rbudget(100).
```

A.1.2 Paths

```
*********************************************************
*  Following is a list of all paths from "s" to "t"     *
*  of the network described in the input data file.     *
*********************************************************

  Path 1: s 1 2 3 t
  Reliability: 0.28


* ------ end ------ *
```

A.1.3 Lower Bound For Original Network

```
LP83 test1.lb output test1.lbr

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Lower Bound Formulation

 ..Objective Maximize

 0.28 f1

 ..Constraints

 Arc 1: f1 <=  100

 Arc 2: f1 <=  50

* ------ end ------ *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  392K bytes.
  Objective Function is MAXIMIZED.
  Variables:         1
  Constraints:       2
     2 LE,   0 EQ,   0 GE.
  Non-zero LP elements:        2
  Disk Space:     0K bytes.
  Page Space:     0K bytes.
  Capacity:  1.9% used.
  Estimated Time: 00:00:00

Iter    1
Solution Time:  00:00:00
U N I Q U E   S O L U T I O N
```

File: Test1 7/24/90 09:44:15 Page 1-1

SOLUTION (Maximized): 14.0000 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	50.0000	0.2800			

File: Test1 7/24/90 09:44:15 Page 1-2

CONSTRAINTS: Lower Bound Formulation

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 1	50.0000 <	100.0000	Arc 2	50.0000 <	50.0000

Total Error: 0.000000

A.1.4 Upper Bound For Original Network

```
LP83 test1.ub output test1.ubr

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Upper Bound Formulation

 ..Objective Maximize

  f1

 ..Constraints

 Arc 1: f1 <=  40.0

 Arc 2: f1 <=  35.0

* ------ end ------ *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  386K bytes.
  Objective Function is MAXIMIZED.
  Variables:          1
  Constraints:        2
     2 LE,    0 EQ,    0 GE.
  Non-zero LP elements:        2
  Disk Space:     OK bytes.
  Page Space:     OK bytes.
  Capacity:  1.9% used.
  Estimated Time: 00:00:00

Iter    1
Solution Time:  00:00:00
U N I Q U E   S O L U T I O N
```

File: Test1 7/24/90 09:51:51 Page 1-1

SOLUTION (Maximized): 35.0000 Upper Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	35.0000	1.0000			

File: Test1 7/24/90 09:51:51 Page 1-2

CONSTRAINTS: Upper Bound Formulation

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 1	35.0000 <	40.0000	Arc 2	35.0000 <	35.0000

Total Error: 0.000000

A.1.5 Investment Strategy Model 1

```
MIP83 output5.lp output test1.is1

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Investment Strategy Model 1

 ..Objective Maximize

 0.28 f1 +
 0 d1 + 0 d2

 ..Constraints

 Arc 1: f1 - d1 <=  100

 Arc 2: f1 - d2 <=  50

 Budget: 5 d1 + 5 d2  <= 100

* ------ end ------ *



Statistics-
  MIP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  371K bytes.
  Objective Function is MAXIMIZED.
  MIP Strategy:       1
  Variables:          3
    Integer:          0
  Constraints:        3
     3 LE,    0 EQ,    0 GE.
  Non-zero LP elements:        6
  Disk Space:     0K bytes.
  Page Space:     0K bytes.
  Capacity:  2.2% used.
  Estimated Time: 00:00:00

Iter    2
Solution Time:  00:00:00
```

U N I Q U E   S O L U T I O N

File: Output5 7/24/90 14:55:17 Page 1-1

SOLUTION (Maximized): 19.6000 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
f1	70.0000	0.2800	d1	0.0000	0.0000
d2	20.0000	0.0000			

File: Output5 7/24/90 14:55:17 Page 1-2

CONSTRAINTS: Investment Strategy Model 1

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 1	70.0000 <	100.0000	Arc 2	50.0000 <	50.0000
Budget	100.0000 <	100.0000			

Total Error: 0.000000

A.1.6 Investment Strategy Model 3

```
MODEL:
MAX= R1 * F1 ;
R1 = ( 0.4 + .1 * X1 ) * ( 0.7 + .1 * X2 ) * ( 1 + .1 * X3 ) ;
0.4 + .1 * X1 < 1 ;
0.7 + .1 * X2 < 1 ;
1 + .1 * X3 < 1 ;
F1 < 100 ;
F1 < 50 ;
5 * X1 + 5 * X2 + 5 * X3  < 100 ;
END
SLB X1 0
SLB X2 0
SLB X3 0
SLB F1 0
LEAVE
```

```
 SOLUTION STATUS:  OPTIMAL TO TOLERANCES.  DUAL CONDITIONS:  SATISFIED.

             OBJECTIVE FUNCTION VALUE

        1)         50.000000

    VARIABLE        VALUE          REDUCED COST
          R1         1.000000           .000000
          F1        50.000000           .000000
          X1         6.000000           .000000
          X2         3.000000           .000000
          X3          .000000           .000000

         ROW   SLACK OR SURPLUS         PRICE
          2)          .000000         50.000000
          3)          .000000         49.999999
          4)          .000000         49.999999
          5)          .000000         49.999999
          6)          .000000          1.000000
          7)         6.000000           .000000
          8)         3.000000           .000000
          9)          .000000           .000000
         10)        55.000000           .000000
```

A.1.7 Output From Maxflo

```
   Original Network

NORMAL STATISTICS
 Mean:   14.0400
 Std. Dev.:   22.4691
 Confidence intvl. (+-):  0.440394

 Reliability:
 0.280800

     Network After Investment Strategy 1

NORMAL STATISTICS
 Mean:   19.656
 Std. Dev.:   31.4589
 Confidence intvl. (+-):  0.616595

 Reliability:
 0.280800

     Network After Investment Strategy 3

NORMAL STATISTICS
 Mean:   50
 Std. Dev.:   0.0
 Confidence intvl. (+-):  0.0

 Reliability:
 1.0
```

A.2 Network 2

A.2.1 Formula Input File

```
%    test2.for

arc(s,1).
arc(1,2).
arc(1,3).
arc(2,4).
arc(3,5).
arc(4,6).
arc(5,7).
arc(6,8).
arc(7,8).
arc(8,t).

prob(s,1).
prob(1,1).
prob(2,1).
prob(3,1).
prob(4,0.5).
prob(5,0.7).
prob(6,0.4).
prob(7,0.6).
prob(8,1).
prob(t,1).

cap(1,*).
cap(2,*).
cap(3,*).
cap(4,400).
cap(5,300).
cap(6,500).
cap(7,300).
cap(8,*).

cost(_,50).

invest(_,0).

budget(1000).

rcost(_,50).

rbudget(1000).
```

A.2.2 Paths

```
**********************************************************
*  Following is a list of all paths from "s" to "t"     *
*  of the network described in the input data file.     *
**********************************************************


  Path 1: s 1 2 4 6 8 t
  Reliability: 0.2

  Path 2: s 1 3 5 7 8 t
  Reliability: 0.42


* ------ end ------ *
```

A.2.3 Lower Bound For Original Network

```
LP83 test2.lb output test2.lbr

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Lower Bound Formulation

 ..Objective Maximize

 0.2 f1 + 0.42 f2

 ..Constraints

 Arc 4: f1 <=  400

 Arc 5: f2 <=  300

 Arc 6: f1 <=  500

 Arc 7: f2 <=  300

* ------ end ------ *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  386K bytes.
  Objective Function is MAXIMIZED.
  Variables:         2
  Constraints:       4
     4 LE,   0 EQ,   0 GE.
  Non-zero LP elements:       4
  Disk Space:     0K bytes.
  Page Space:     0K bytes.
  Capacity:  2.2% used.
  Estimated Time: 00:00:00

Iter    2
Solution Time:  00:00:00
*May have* A L T E R N A T E   S O L U T I O N
```

File: Test2 7/24/90 10:40:43 Page 1-1

SOLUTION (Maximized): 206.0000 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	400.0000	0.2000	f2	300.0000	0.4200

File: Test2 7/24/90 10:40:43 Page 1-2

CONSTRAINTS: Lower Bound Formulation

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 4	400.0000 <	400.0000	Arc 5	300.0000 <	300.0000
Arc 6	400.0000 <	500.0000	Arc 7	300.0000 <	300.0000

Total Error: 0.000000

A.2.4 Upper Bound For Original Network

```
LP83 test2.ub output test2.ubr

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Upper Bound Formulation

 ..Objective Maximize

  f1 +  f2

 ..Constraints

 Arc 4: f1 <=  200.0

 Arc 5: f2 <=  210.0

 Arc 6: f1 <=  200.0

 Arc 7: f2 <=  180.0

* ------ end ------- *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  386K bytes.
  Objective Function is MAXIMIZED.
  Variables:         2
  Constraints:       4
     4 LE,   0 EQ,   0 GE.
  Non-zero LP elements:       4
  Disk Space:     0K bytes.
  Page Space:     0K bytes.
  Capacity:  2.2% used.
  Estimated Time: 00:00:00

Iter    2
Solution Time:  00:00:00
*May have* A L T E R N A T E   S O L U T I O N
```

File: Test2 7/24/90 10:41:10 Page 1-1

SOLUTION (Maximized): 380.0000 Upper Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	200.0000	1.0000	f2	180.0000	1.0000

File: Test2 7/24/90 10:41:10 Page 1-2

CONSTRAINTS: Upper Bound Formulation

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 4	200.0000 <	200.0000	Arc 5	180.0000 <	210.0000
Arc 6	200.0000 <	200.0000	Arc 7	180.0000 <	180.0000

Total Error: 0.000000

A.2.5 Investment Strategy Model 1

```
MIP83 output5.lp output test2.is1

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Investment Strategy Model 1

 ..Objective Maximize

 0.2 f1 + 0.42 f2 +
 0 d4 + 0 d5 + 0 d6 + 0 d7

 ..Constraints

 Arc 4: f1 - d4 <=  400

 Arc 5: f2 - d5 <=  300

 Arc 6: f1 - d6 <=  500

 Arc 7: f2 - d7 <=  300

 Budget:- 50 d4 + 50 d5 + 50 d6 + 50 d7  <= 1000

* ------ end ------ *



Statistics-
  MIP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  371K bytes.
  Objective Function is MAXIMIZED.
  MIP Strategy:       1
  Variables:          6
    Integer:          0
  Constraints:        5
     5 LE,   0 EQ,    0 GE.
  Non-zero LP elements:      12
  Disk Space:     0K bytes.
  Page Space:     0K bytes.
  Capacity:  2.7% used.
```

Estimated Time: 00:00:00

Iter 4
Solution Time: 00:00:00
U N I Q U E   S O L U T I O N

File: Output5 7/25/90 16:22:20 Page 1-1
SOLUTION (Maximized): 210.2000 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
f1	400.0000	0.2000	f2	310.0000	0.4200
d4	0.0000	0.0000	d5	10.0000	0.0000
d6	0.0000	0.0000	d7	10.0000	0.0000

File: Output5 7/25/90 16:22:20 Page 1-2

CONSTRAINTS: Investment Strategy Model 1

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 4	400.0000	<	400.0000	Arc 5	300.0000	<	300.0000
Arc 6	400.0000	<	500.0000	Arc 7	300.0000	<	300.0000
Budget	1,000.0000	<	1,000.0000				

Total Error: 0.000000

A.2.6 Investment Strategy Model 3

```
MODEL:
MAX= R1 * F1 + R2 * F2 ;
R1 = ( 1 + .1 * X1 ) * ( 1 + .1 * X2 ) * ( 0.5 + .1 * X4 ) *
     ( 0.4 + .1 * X6 ) * ( 1 + .1 * X8 ) ;
R2 = ( 1 + .1 * X1 ) * ( 1 + .1 * X3 ) * ( 0.7 + .1 * X5 ) *
     ( 0.6 + .1 * X7 ) * ( 1 + .1 * X8 ) ;
1 + .1 * X1 < 1 ;
1 + .1 * X2 < 1 ;
1 + .1 * X3 < 1 ;
0.5 + .1 * X4 < 1 ;
0.7 + .1 * X5 < 1 ;
0.4 + .1 * X6 < 1 ;
0.6 + .1 * X7 < 1 ;
1 + .1 * X8 < 1 ;
F1 < 400 ;
F2 < 300 ;
F1 < 500 ;
F2 < 300 ;
5 * X1 + 5 * X2 + 5 * X3 + 5 * X4 + 5 * X5 +
5 * X6 + 5 * X7 + 5 * X8  < 100 ;
END
SLB X1 0
SLB X2 0
SLB X3 0
SLB X4 0
SLB X5 0
SLB X6 0
SLB X7 0
SLB X8 0
SLB F1 0
SLB F2 0
LEAVE

 SOLUTION STATUS:  OPTIMAL TO TOLERANCES.  DUAL CONDITIONS:  SATISFIED.

            OBJECTIVE FUNCTION VALUE

        1)       700.000000

    VARIABLE        VALUE          REDUCED COST
          R1         1.000000           .000000
          F1       400.000000           .000000
          R2         1.000000           .000000
          F2       300.000000           .000000
          X1          .000000           .000000
          X2          .000000           .000000
          X4         5.000000           .000000
          X6         6.000000           .000000
          X8          .000000           .000000
```

X3	.000000	.000000
X5	3.000000	.000000
X7	4.000000	.000000

ROW	SLACK OR SURPLUS	PRICE
2)	.000000	400.000000
3)	.000000	300.000000
4)	.000000	699.999990
5)	.000000	399.999994
6)	.000000	299.999996
7)	.000000	399.999994
8)	.000000	299.999996
9)	.000000	399.999994
10)	.000000	299.999996
11)	.000000	699.999990
12)	.000000	1.000000
13)	.000000	.000000
14)	100.000000	.000000
15)	.000000	1.000000
16)	100.000022	.000000

A.2.7 Output From Maxflo

```
   Original Network

NORMAL STATISTICS
 Mean:   209.550
 Std. Dev.:   220.964
 Confidence intvl. (+-):  4.33089

 Reliability:
 0.540200

     Network After Investment Strategy 1

NORMAL STATISTICS
 Mean:   213.827
 Std. Dev.:  224.389
 Confidence intvl. (+-):  4.39803

 Reliability:
 0.540200

     Network After Investment Strategy 3

NORMAL STATISTICS
 Mean:   700.000
 Std. Dev.:   0.0
 Confidence intvl. (+-):  0.0

 Reliability:
 1.0
```

A.3 Network 3

A.3.1 Formula Input File

```
% test3.for
arc(s,1).
arc(s,2).
arc(1,3).
arc(1,4).
arc(1,5).
arc(2,6).
arc(3,6).
arc(4,8).
arc(5,7).
arc(5,9).
arc(6,7).
arc(6,9).
arc(7,8).
arc(8,t).
arc(9,t).

prob(s,1).
prob(1,1).
prob(2,0.1).
prob(3,0).
prob(4,1).
prob(5,1).
prob(6,0.5).
prob(7,0).
prob(8,0.9).
prob(9,1).
prob(t,1).

cap(1,5).
cap(2,6).
cap(3,5).
cap(4,2).
cap(5,4).
cap(6,5).
cap(7,5).
cap(8,4).
cap(9,7).

cost(_,5).

invest(_,0).

budget(100).

rcost(_,5).

rbudget(100).
```

A.3.2 Paths

```
*****************************************************
*  Following is a list of all paths from "s" to "t"    *
*  of the network described in the input data file.    *
*****************************************************


  Path 1: s 1 3 6 7 8 t
  Reliability: 0.0

  Path 2: s 1 3 6 9 t
  Reliability: 0.0

  Path 3: s 1 4 8 t
  Reliability: 0.9

  Path 4: s 1 5 7 8 t
  Reliability: 0.0

  Path 5: s 1 5 9 t
  Reliability: 1

  Path 6: s 2 6 7 8 t
  Reliability: 0.0

  Path 7: s 2 6 9 t
  Reliability: 0.05


* ------ end ------ *
```

A.3.3 Lower Bound For Original Network

```
LP83 output3.lp output test3.lbr

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Lower Bound Formulation

 ..Objective Maximize

 0.0 f1 + 0.0 f2 + 0.9 f3 + 0.0 f4 +
 1 f5 + 0.0 f6 + 0.05 f7

 ..Constraints

 Arc 1: f1 + f2 + f3 + f4 + f5 <=  5

 Arc 2: f6 + f7 <=  6

 Arc 3: f1 + f2 <=  5

 Arc 4: f3 <=  2

 Arc 5: f4 + f5 <=  4

 Arc 6: f1 + f2 + f6 + f7 <=  5

 Arc 7: f1 + f4 + f6 <=  5

 Arc 8: f1 + f3 + f4 + f6 <=  4

 Arc 9: f2 + f5 + f7 <=  7

* ------ end ------ *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  392K bytes.
  Objective Function is MAXIMIZED.
  Variables:         7
```

```
  Constraints:      9
     9 LE,    0 EQ,    0 GE.
  Non-zero LP elements:     26
  Disk Space:     0K bytes.
  Page Space:     1K bytes.
  Capacity:  3.2% used.
  Estimated Time: 00:00:00

Iter    3
Solution Time:  00:00:00
A L T E R N A T E   S O L U T I O N S
```

File: Output3 7/24/90 12:18:42 Page 1-1

SOLUTION (Maximized): 5.0500 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	0.0000	f2	0.0000	0.0000
f3	1.0000	0.9000	f4	0.0000	0.0000
f5	4.0000	1.0000	f6	0.0000	0.0000
f7	3.0000	0.0500			

File: Output3 7/24/90 12:18:42 Page 1-2

CONSTRAINTS: Lower Bound Formulation

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 1	5.0000 <	5.0000	Arc 2	3.0000 <	6.0000
Arc 3	0.0000 <	5.0000	Arc 4	1.0000 <	2.0000
Arc 5	4.0000 <	4.0000	Arc 6	3.0000 <	5.0000
Arc 7	0.0000 <	5.0000	Arc 8	1.0000 <	4.0000
Arc 9	7.0000 <	7.0000			

Total Error: 0.000000

File: Output3 7/24/90 12:18:42 Page 2-1

SOLUTION (Maximized): 5.0500 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	0.0000	f2	0.0000	0.0000
f3	1.0000	0.9000	f4	0.0000	0.0000
f5	4.0000	1.0000	f6	2.0000	0.0000
f7	3.0000	0.0500			

File: Output3 7/24/90 12:18:42 Page 2-2

CONSTRAINTS: Lower Bound Formulation

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 1	5.0000	<	5.0000	Arc 2	5.0000	<	6.0000
Arc 3	0.0000	<	5.0000	Arc 4	1.0000	<	2.0000
Arc 5	4.0000	<	4.0000	Arc 6	5.0000	<	5.0000
Arc 7	2.0000	<	5.0000	Arc 8	3.0000	<	4.0000
Arc 9	7.0000	<	7.0000				

Total Error: 0.000000

File: Output3 7/24/90 12:18:42 Page 3-1

SOLUTION (Maximized): 5.0500 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	0.0000	f2	0.0000	0.0000
f3	1.0000	0.9000	f4	0.0000	0.0000
f5	4.0000	1.0000	f6	0.0000	0.0000
f7	3.0000	0.0500			

CONSTRAINTS: Lower Bound Formulation

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 1	5.0000	<	5.0000	Arc 2	3.0000	<	6.0000
Arc 3	0.0000	<	5.0000	Arc 4	1.0000	<	2.0000
Arc 5	4.0000	<	4.0000	Arc 6	3.0000	<	5.0000
Arc 7	0.0000	<	5.0000	Arc 8	1.0000	<	4.0000
Arc 9	7.0000	<	7.0000				

Total Error: 0.000000

A.3.4 Upper Bound For Original Network

```
LP83 output4.lp output test3.ubr

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Upper Bound Formulation

 ..Objective Maximize

  f1 +  f2 +  f3 +  f4 +
  f5 +  f6 +  f7

 ..Constraints

 Arc 1: f1 + f2 + f3 + f4 + f5 <=  5

 Arc 2: f6 + f7 <=  0.6

 Arc 3: f1 + f2 <=  0

 Arc 4: f3 <=  2

 Arc 5: f4 + f5 <=  4

 Arc 6: f1 + f2 + f6 + f7 <=  2.5

 Arc 7: f1 + f4 + f6 <=  0

 Arc 8: f1 + f3 + f4 + f6 <=  3.6

 Arc 9: f2 + f5 + f7 <=  7

* ------ end ------ *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  392K bytes.
  Objective Function is MAXIMIZED.
  Variables:        7
```

```
Constraints:       9
   9 LE,    0 EQ,    0 GE.
Non-zero LP elements:      26
Disk Space:      0K bytes.
Page Space:      1K bytes.
Capacity:  3.2% used.
Estimated Time: 00:00:00

Iter    3
Solution Time:  00:00:0
A L T E R N A T E   S O L U T I O N S
```

File: Output4 10/02/90 11:33:58 Page 1-1

SOLUTION (Maximized): 5.6000 Upper Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	1.0000	f2	0.0000	1.0000
f3	2.0000	1.0000	f4	0.0000	1.0000
f5	3.0000	1.0000	f6	0.0000	1.0000
f7	0.6000	1.0000			

CONSTRAINTS: Upper Bound Formulation

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 1	5.0000	<	5.0000	Arc 2	0.6000	<	0.6000
Arc 3	0.0000	<	0.0000	Arc 4	2.0000	<	2.0000
Arc 5	3.0000	<	4.0000	Arc 6	0.6000	<	2.5000
Arc 7	0.0000	<	0.0000	Arc 8	2.0000	<	3.6000
Arc 9	3.6000	<	7.0000				

Total Error: 0.000000

A.3.5 Investment Strategy Model 1

```
LP83 output5.lp output test3.is1

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Investment Strategy Model 1

 ..Objective Maximize

 0.0 f1 + 0.0 f2 + 0.9 f3 + 0.0 f4 +
 1 f5 + 0.0 f6 + 0.05 f7 +
 0 d1 + 0 d2 + 0 d3 + 0 d4 + 0 d5 + 0 d6 + 0 d7 +
 0 d8 + 0 d9

 ..Constraints

 Arc 1: f1 + f2 + f3 + f4 + f5 - d1 <=  5

 Arc 2: f6 + f7 - d2 <=  6

 Arc 3: f1 + f2 - d3 <=  5

 Arc 4: f3 - d4 <=  2

 Arc 5: f4 + f5 - d5 <=  4

 Arc 6: f1 + f2 + f6 + f7 - d6 <=  5

 Arc 7: f1 + f4 + f6 - d7 <=  5

 Arc 8: f1 + f3 + f4 + f6 - d8 <=  4

 Arc 9: f2 + f5 + f7 - d9 <=  7

 Budget: 5 d1 + 5 d2 + 5 d3 + 5 d4 + 5 d5 +
         5 d6 + 5 d7 + 5 d8 + 5 d9  <= 100

* ------ end ------ *



Statistics-
```

```
MIP83 Version 5.00a
Machine memory:  640K bytes.
Pagable memory:  371K bytes.
Objective Function is MAXIMIZED.
MIP Strategy:       1
Variables:         16
  Integer:          0
Constraints:       10
  10 LE,    0 EQ,    0 GE.
Non-zero LP elements:      44
Disk Space:      0K bytes.
Page Space:      1K bytes.
Capacity:  4.1% used.
Estimated Time: 00:00:01

Iter    6
Solution Time:  00:00:01
U N I Q U E   S O L U T I O N
```

File: Output5 7/25/90 16:23:57 Page 1-1

SOLUTION (Maximized): 13.6000 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	0.0000	f2	0.0000	0.0000
f3	4.0000	0.9000	f4	0.0000	0.0000
f5	10.0000	1.0000	f6	0.0000	0.0000
f7	0.0000	0.0500	d1	9.0000	0.0000
d2	0.0000	0.0000	d3	0.0000	0.0000
d4	2.0000	0.0000	d5	6.0000	0.0000
d6	0.0000	0.0000	d7	0.0000	0.0000
d8	0.0000	0.0000	d9	3.0000	0.0000

CONSTRAINTS: Investment Strategy Model 1

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 1	5.0000 <	5.0000	Arc 2	0.0000 <	6.0000
Arc 3	0.0000 <	5.0000	Arc 4	2.0000 <	2.0000
Arc 5	4.0000 <	4.0000	Arc 6	0.0000 <	5.0000
Arc 7	0.0000 <	5.0000	Arc 8	4.0000 <	4.0000
Arc 9	7.0000 <	7.0000	Budget	100.0000 <	100.0000

Total Error: 0.000000

A.3.6 Investment Strategy Model 3

```
MODEL:
MAX= R1 * F1 + R2 * F2 + R3 * F3 + R4 * F4 + R5 * F5 +
     R6 * F6 + R7 * F7 ;
R1 = ( 1 + .1 * X1 ) * ( 0 + .1 * X3 ) * ( 0.5 + .1 * X6 ) *
     ( 0 + .1 * X7 ) * ( 0.9 + .1 * X8 ) ;
R2 = ( 1 + .1 * X1 ) * ( 0 + .1 * X3 ) * ( 0.5 + .1 * X6 ) *
     ( 1 + .1 * X9 ) ;
R3 = ( 1 + .1 * X1 ) * ( 1 + .1 * X4 ) * ( 0.9 + .1 * X8 ) ;
R4 = ( 1 + .1 * X1 ) * ( 1 + .1 * X5 ) * ( 0 + .1 * X7 ) *
     ( 0.9 + .1 * X8 ) ;
R5 = ( 1 + .1 * X1 ) * ( 1 + .1 * X5 ) * ( 1 + .1 * X9 ) ;
R6 = ( 0.1 + .1 * X2 ) * ( 0.5 + .1 * X6 ) * ( 0 + .1 * X7 ) *
     ( 0.9 + .1 * X8 ) ;
R7 = ( 0.1 + .1 * X2 ) * ( 0.5 + .1 * X6 ) * ( 1 + .1 * X9 ) ;
1 + .1 * X1 < 1 ;
0.1 + .1 * X2 < 1 ;
0 + .1 * X3 < 1 ;
1 + .1 * X4 < 1 ;
1 + .1 * X5 < 1 ;
0.5 + .1 * X6 < 1 ;
0 + .1 * X7 < 1 ;
0.9 + .1 * X8 < 1 ;
1 + .1 * X9 < 1 ;
F1 + F2 + F3 + F4 + F5 < 5 ;
F6 + F7 < 6 ;
F1 + F2 < 5 ;
F3 < 2 ;
F4 + F5 < 4 ;
F1 + F2 + F6 + F7 < 5 ;
F1 + F4 + F6 < 5 ;
F1 + F3 + F4 + F6 < 4 ;
F2 + F5 + F7 < 7 ;
5 * X1 + 5 * X2 + 5 * X3 + 5 * X4 + 5 * X5 +
5 * X6 + 5 * X7 + 5 * X8 + 5 * X9  < 100 ;
END
SLB X1 0
SLB X2 0
SLB X3 0
SLB X4 0
SLB X5 0
SLB X6 0
SLB X7 0
SLB X8 0
SLB X9 0
SLB F1 0
SLB F2 0
SLB F3 0
SLB F4 0
SLB F5 0
```

```
SLB F6 0
SLB F7 0
LEAVE
```

SOLUTION STATUS: OPTIMAL TO TOLERANCES. DUAL CONDITIONS: SATISFIED.

OBJECTIVE FUNCTION VALUE

1)	9.500000

VARIABLE	VALUE	REDUCED COST
R1	0.000000	0.000000
F1	0.000000	0.000000
R2	0.000000	0.000000
F2	0.000000	0.000000
R3	1.000000	0.000000
F3	2.000000	0.000000
R4	0.500000	0.000000
F4	0.000000	0.000000
R5	1.000000	0.000000
F5	3.000000	0.000000
R6	0.500000	0.000000
F6	1.000000	0.000000
R7	1.000000	0.000000
F7	4.000000	0.000000
X1	0.000000	0.000000
X3	0.000000	0.000000
X6	5.000000	0.000000
X7	5.000000	0.000000
X8	1.000000	0.000000
X9	0.000000	0.000000
X4	0.000000	0.000000
X5	0.000000	0.000000
X2	9.000000	0.000000

ROW	SLACK OR SURPLUS	PRICE
2)	0.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	2.000000
5)	0.000000	0.000000
6)	0.000000	3.000000
7)	0.000000	1.000000
8)	0.000000	4.000000
9)	0.000000	4.000000
10)	0.000000	3.500000
11)	1.000000	0.000000
12)	0.000000	1.000000
13)	0.000000	2.000000
14)	0.000000	3.500000
15)	0.500000	0.000000

16)	0.000000	1.500000
17)	0.000000	6.000000
18)	0.000000	0.500000
19)	1.000000	0.000000
20)	5.000000	0.000000
21)	0.000000	0.500000
22)	1.000000	0.000000
23)	0.000000	0.500000
24)	4.000000	0.000000
25)	1.000000	0.000000
26)	0.000000	0.500000
27)	0.000000	0.000000
28)	9.000000	0.000000
29)	0.000000	-0.100000
30)	0.000000	0.000000
31)	0.000000	0.000000
32)	5.000000	0.000000
33)	5.000000	0.000000
34)	1.000000	0.000000
35)	0.000000	0.000000
36)	0.000000	-1.000000
37)	0.000000	-1.500000
38)	2.000000	0.000000
39)	0.000000	0.000000
40)	3.000000	0.000000
41)	1.000000	0.000000
42)	4.000000	0.000000
43)	0.000000	0.020000

A.3.7 Output From Maxflo

```
   Original Network

NORMAL STATISTICS
 Mean:   5.08490
 Std. Dev.:  0.900810
 Confidence intvl. (+-):  1.76559e-02

 Reliability:
 1.00000

     Network After Investment Strategy 1

NORMAL STATISTICS
 Mean:   13.5816
 Std. Dev.:  1.22419
 Confidence intvl. (+-):  2.39942e-02

 Reliability:
 1.00000

     Network After Investment Strategy 3

NORMAL STATISTICS
 Mean:   9.49650
 Std. Dev.: 0.500009
 Confidence intvl. (+-):  9.80018e-03

 Reliability:
 1.00000
```

A.4 Network 4

A.4.1 Formula Input File

```
% test4.for

arc(s,1).
arc(1,2).
arc(2,3).
arc(2,4).
arc(2,5).
arc(2,6).
arc(3,7).
arc(4,15).
arc(5,9).
arc(6,11).
arc(7,8).
arc(7,12).
arc(8,9).
arc(9,10).
arc(9,13).
arc(10,11).
arc(11,14).
arc(12,15).
arc(13,15).
arc(14,17).
arc(15,16).
arc(16,17).
arc(17,18).
arc(18,t).

prob(s,1).
prob(1,1).
prob(2,1).
prob(3,1).
prob(4,0).
prob(5,1).
prob(6,1).
prob(7,0.3).
prob(8,0).
prob(9,0.7).
prob(10,0).
prob(11,0.5).
prob(12,0.6).
prob(13,0.3).
prob(14,0.6).
prob(15,0.8).
prob(16,0.7).
prob(17,1).
prob(18,1).
prob(t,1).
```

```
cap(1,*).
cap(2,1000).
cap(3,500).
cap(4,5).
cap(5,300).
cap(6,600).
cap(7,500).
cap(8,5).
cap(9,300).
cap(10,5).
cap(11,300).
cap(12,500).
cap(13,200).
cap(14,300).
cap(15,600).
cap(16,350).
cap(17,*).
cap(18,*).

cost(_,50).

invest(_,0).

budget(1000).

rcost(_,50).

rbudget(1000).
```

A.4.2 Paths

```
*******************************************************
* Following is a list of all paths from "s" to "t"    *
* of the network described in the input data file.    *
*******************************************************


  Path 1: s 1 2 3 7 8 9 10 11 14 17 18 t
  Reliability: 0.0

  Path 2: s 1 2 3 7 8 9 13 15 16 17 18 t
  Reliability: 0.0

  Path 3: s 1 2 3 7 12 15 16 17 18 t
  Reliability: 0.1008

  Path 4: s 1 2 4 15 16 17 18 t
  Reliability: 0.0

  Path 5: s 1 2 5 9 10 11 14 17 18 t
  Reliability: 0.0

  Path 6: s 1 2 5 9 13 15 16 17 18 t
  Reliability: 0.1176

  Path 7: s 1 2 6 11 14 17 18 t
  Reliability: 0.3


* ------ end ------ *
```

A.4.3 Lower Bound For Original Network

```
LP83 output3.lp output test4.lbr

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Lower Bound Formulation

 ..Objective Maximize

 0.0 f1 + 0.0 f2 + 0.1008 f3 + 0.0 f4 +
 0.0 f5 + 0.1176 f6 + 0.3 f7

 ..Constraints

 Arc 2: f1 + f2 + f3 + f4 + f5 + f6 + f7 <=  1000

 Arc 3: f1 + f2 + f3 <=  500

 Arc 4: f4 <=  5

 Arc 5: f5 + f6 <=  300

 Arc 6: f7 <=  600

 Arc 7: f1 + f2 + f3 <=  500

 Arc 8: f1 + f2 <=  5

 Arc 9: f1 + f2 + f5 + f6 <=  300

 Arc 10: f1 + f5 <=  5

 Arc 11: f1 + f5 + f7 <=  300

 Arc 12: f3 <=  500

 Arc 13: f2 + f6 <=  200

 Arc 14: f1 + f5 + f7 <=  300

 Arc 15: f2 + f3 + f4 + f6 <=  600

 Arc 16: f2 + f3 + f4 + f6 <=  350
```

```
* ------ end ------ *


Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  392K bytes.
  Objective Function is MAXIMIZED.
  Variables:          7
  Constraints:       15
    15 LE,    0 EQ,    0 GE.
  Non-zero LP elements:       42
  Disk Space:      0K bytes.
  Page Space:      1K bytes.
  Capacity:  3.9% used.
  Estimated Time: 00:00:01

Iter    3
Solution Time:  00:00:00
*May have* A L T E R N A T E   S O L U T I O N
```

File: Output3 10/03/90 07:45:27 Page 1-1

SOLUTION (Maximized): 128.6400 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	0.0000	f2	0.0000	0.0000
f3	150.0000	0.1008	f4	0.0000	0.0000
f5	0.0000	0.0000	f6	200.0000	0.1176
f7	300.0000	0.3000			

File: Output3 10/03/90 07:45:27 Page 1-2

CONSTRAINTS: Lower Bound Formulation

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 2	650.0000 <	1,000.0000	Arc 3	150.0000 <	500.0000
Arc 4	0.0000 <	5.0000	Arc 5	200.0000 <	300.0000
Arc 6	300.0000 <	600.0000	Arc 7	150.0000 <	500.0000
Arc 8	0.0000 <	5.0000	Arc 9	200.0000 <	300.0000
Arc 10	0.0000 <	5.0000	Arc 11	300.0000 <	300.0000
Arc 12	150.0000 <	500.0000	Arc 13	200.0000 <	200.0000
Arc 14	300.0000 <	300.0000	Arc 15	350.0000 <	600.0000
Arc 16	350.0000 <	350.0000			

Total Error: 0.000000

A.4.4 Upper Bound For Original Network

```
LP83 output4.lp output test4.ubr

 Copyright (C) 1986 by Sunset Software.
 All Rights Reserved Worldwide.
 1613 Chelsea Road,  Suite 153
 San Marino, California  91108   U.S.A.

 Licensed Solely To: Wright-Patterson Air Force Base

..Title

Upper Bound Formulation

..Objective Maximize

 f1 +  f2 +  f3 +  f4 +
 f5 +  f6 +  f7

..Constraints

Arc 2: f1 + f2 + f3 + f4 + f5 + f6 + f7 <=  1000

Arc 3: f1 + f2 + f3 <=  500

Arc 4: f4 <=  0

Arc 5: f5 + f6 <=  300

Arc 6: f7 <=  600

Arc 7: f1 + f2 + f3 <=  150.0

Arc 8: f1 + f2 <=  0

Arc 9: f1 + f2 + f5 + f6 <=  210.0

Arc 10: f1 + f5 <=  0

Arc 11: f1 + f5 + f7 <=  150.0

Arc 12: f3 <=  300.0

Arc 13: f2 + f6 <=  60.0

Arc 14: f1 + f5 + f7 <=  180.0

Arc 15: f2 + f3 + f4 + f6 <=  480.0

Arc 16: f2 + f3 + f4 + f6 <=  245.0
```

```
* ------ end ------ *


Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  392K bytes.
  Objective Function is MAXIMIZED.
  Variables:         7
  Constraints:       15
    15 LE,    0 EQ,    0 GE.
  Non-zero LP elements:       42
  Disk Space:      0K bytes.
  Page Space:      1K bytes.
  Capacity:  3.9% used.
  Estimated Time: 00:00:01

Iter    6
Solution Time:  00:00:01
A L T E R N A T E   S O L U T I O N S


File: Output4                                    7/24/90 13:14:48 Page 1-1
SOLUTION (Maximized):     360.0000    Upper Bound Formulation
```

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	1.0000	f2	0.0000	1.0000
f3	150.0000	1.0000	f4	0.0000	1.0000
f5	0.0000	1.0000	f6	60.0000	1.0000
f7	150.0000	1.0000			

File: Output4 7/24/90 13:14:48 Page 1-2

CONSTRAINTS: Upper Bound Formulation

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 2	360.0000 <	1,000.0000	Arc 3	150.0000 <	500.0000
Arc 4	0.0000 <	0.0000	Arc 5	60.0000 <	300.0000
Arc 6	150.0000 <	600.0000	Arc 7	150.0000 <	150.0000
Arc 8	0.0000 <	0.0000	Arc 9	60.0000 <	210.0000
Arc 10	0.0000 <	0.0000	Arc 11	150.0000 <	150.0000
Arc 12	150.0000 <	300.0000	Arc 13	60.0000 <	60.0000
Arc 14	150.0000 <	180.0000	Arc 15	210.0000 <	480.0000
Arc 16	210.0000 <	245.0000			

Total Error: 0.000000

A.4.5 Investment Strategy Model 1

```
LP83 a:output5 output a:test4.is1

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Investment Strategy Model 1

 ..Objective Maximize

 0.0 f1 + 0.0 f2 + 0.1008 f3 + 0.0 f4 + 0.0 f5 +
 0.1176 f6 + 0.3 f7 + 0 d2 + 0 d3 + 0 d4 + 0 d5 +
 0 d6 + 0 d7 + 0 d8 + 0 d9 + 0 d10 +
 0 d11 + 0 d12 + 0 d13 + 0 d14 + 0 d15 + 0 d16

 ..Constraints

 Arc 2: f1 + f2 + f3 + f4 + f5 + f6 + f7 - d2 <=  1000

 Arc 3: f1 + f2 + f3 - d3 <=  500

 Arc 4: f4 - d4 <=  5

 Arc 5: f5 + f6 - d5 <=  300

 Arc 6: f7 - d6 <=  600

 Arc 7: f1 + f2 + f3 - d7 <=  500

 Arc 8: f1 + f2 - d8 <=  5

 Arc 9: f1 + f2 + f5 + f6 - d9 <=  300

 Arc 10: f1 + f5 - d10 <=  5

 Arc 11: f1 + f5 + f7 - d11 <=  300

 Arc 12: f3 - d12 <=  500

 Arc 13: f2 + f6 - d13 <=  200

 Arc 14: f1 + f5 + f7 - d14 <=  300

 Arc 15: f2 + f3 + f4 + f6 - d15 <=  600
```

Arc 16: f2 + f3 + f4 + f6 - d16 <= 350

Budget: 50 d2 + 50 d3 + 50 d4 + 50 d5 +
50 d6 + 50 d7 + 50 d8 + d9 + 50 d10 +
50 d11 + 50 d12 + 50 d13 + 50 d14 + 50 d15 +
50 d16 <= 1000

Statistics-
LP83 Version 5.00a
Machine memory: 624K bytes.
Pagable memory: 393K bytes.
Objective Function is MAXIMIZED.
Variables: 22
Constraints: 16
16 LE, 0 EQ, 0 GE.
Non-zero LP elements: 72
Disk Space: 0K bytes.
Page Space: 3K bytes.
Capacity: 5.2% used.
Estimated Time: 00:00:04

Iter 5
Solution Time: 00:00:01
U N I Q U E S O L U T I O N

File: output5 7/26/90 13:48:49 Page 1-1
SOLUTION (Maximized): 131.6400 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	0.0000	f2	0.0000	0.0000
f3	150.0000	0.1008	f4	0.0000	0.0000
f5	0.0000	0.0000	f6	200.0000	0.1176
f7	310.0000	0.3000	d2	0.0000	0.0000
d3	0.0000	0.0000	d4	0.0000	0.0000
d5	0.0000	0.0000	d6	0.0000	0.0000
d7	0.0000	0.0000	d8	0.0000	0.0000
d9	0.0000	0.0000	d10	0.0000	0.0000
d11	10.0000	0.0000	d12	0.0000	0.0000

d13	0.0000	0.0000	d14	10.0000	0.0000

File: output5 7/26/90 13:48:49 Page 1-2

SOLUTION (Maximized): 131.6400 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
d15	0.0000	0.0000	d16	0.0000	0.0000

File: output5 7/26/90 13:48:49 Page 1-3

CONSTRAINTS: Investment Strategy Model 1

Constraint	Activity	RHS	Constraint	Activity	RHS
Arc 2	660.0000 <	1,000.0000	Arc 3	150.0000 <	500.0000
Arc 4	0.0000 <	5.0000	Arc 5	200.0000 <	300.0000
Arc 6	310.0000 <	600.0000	Arc 7	150.0000 <	500.0000
Arc 8	0.0000 <	5.0000	Arc 9	200.0000 <	300.0000
Arc 10	0.0000 <	5.0000	Arc 11	300.0000 <	300.0000
Arc 12	150.0000 <	500.0000	Arc 13	200.0000 <	200.0000
Arc 14	300.0000 <	300.0000	Arc 15	350.0000 <	600.0000
Arc 16	350.0000 <	350.0000	Budget	1,000.0000 <	1,000.0000

Total Error: 0.000000

A.4.6 Investment Strategy Model 3

```
MODEL:
MAX= R1 * F1 + R2 * F2 + R3 * F3 + R4 * F4 + R5 * F5 +
     R6 * F6 + R7 * F7 ;
R1 = ( 1 + .1 * X1 ) * ( 1 + .1 * X2 ) * ( 1 + .1 * X3 ) *
     ( 0.3 + .1 * X7 ) * ( 0 + .1 * X8 ) * ( 0.7 + .1 * X9 ) *
     ( 0 + .1 * X10 ) * ( 0.5 + .1 * X11 ) * ( 0.6 + .1 * X14 ) *
     ( 1 + .1 * X17 ) * ( 1 + .1 * X18 ) ;
R2 = ( 1 + .1 * X1 ) * ( 1 + .1 * X2 ) * ( 1 + .1 * X3 ) *
     ( 0.3 + .1 * X7 ) * ( 0 + .1 * X8 ) * ( 0.7 + .1 * X9 ) *
     ( 0.3 + .1 * X13 ) * ( 0.8 + .1 * X15 ) * ( 0.7 + .1 * X16 ) *
     ( 1 + .1 * X17 ) * ( 1 + .1 * X18 ) ;
R3 = ( 1 + .1 * X1 ) * ( 1 + .1 * X2 ) * ( 1 + .1 * X3 ) *
     ( 0.3 + .1 * X7 ) * ( 0.6 + .1 * X12 ) * ( 0.8 + .1 * X15 ) *
     ( 0.7 + .1 * X16 ) * ( 1 + .1 * X17 ) * ( 1 + .1 * X18 ) ;
R4 = ( 1 + .1 * X1 ) * ( 1 + .1 * X2 ) * ( 0 + .1 * X4 ) *
     ( 0.8 + .1 * X15 ) * ( 0.7 + .1 * X16 ) * ( 1 + .1 * X17 ) *
     ( 1 + .1 * X18 ) ;
R5 = ( 1 + .1 * X1 ) * ( 1 + .1 * X2 ) * ( 1 + .1 * X5 ) *
     ( 0.7 + .1 * X9 ) * ( 0 + .1 * X10 ) * ( 0.5 + .1 * X11 ) *
     ( 0.6 + .1 * X14 ) * ( 1 + .1 * X17 ) * ( 1 + .1 * X18 ) ;
R6 = ( 1 + .1 * X1 ) * ( 1 + .1 * X2 ) * ( 1 + .1 * X5 ) *
     ( 0.7 + .1 * X9 ) * ( 0.3 + .1 * X13 ) * ( 0.8 + .1 * X15 ) *
R7 = ( 1 + .1 * X1 ) * ( 1 + .1 * X2 ) * ( 1 + .1 * X6 ) *
     ( 0.5 + .1 * X11 ) * ( 0.6 + .1 * X14 ) * ( 1 + .1 * X17 ) *
     ( 1 + .1 * X18 ) ;
1 + .1 * X1 < 1 ;
1 + .1 * X2 < 1 ;
1 + .1 * X3 < 1 ;
0 + .1 * X4 < 1 ;
1 + .1 * X5 < 1 ;
1 + .1 * X6 < 1 ;
0.3 + .1 * X7 < 1 ;
0 + .1 * X8 < 1 ;
0.7 + .1 * X9 < 1 ;
0 + .1 * X10 < 1 ;
0.5 + .1 * X11 < 1 ;
0.6 + .1 * X12 < 1 ;
0.3 + .1 * X13 < 1 ;
0.6 + .1 * X14 < 1 ;
0.8 + .1 * X15 < 1 ;
0.7 + .1 * X16 < 1 ;
1 + .1 * X17 < 1 ;
1 + .1 * X18 < 1 ;
F1 + F2 + F3 + F4 + F5 + F6 + F7 < 1000 ;
F1 + F2 + F3 < 500 ;
F4 < 5 ;
F5 + F6 < 300 ;
F7 < 600 ;
F1 + F2 + F3 < 500 ;
```

```
F1 + F2 < 5 ;
F1 + F2 + F5 + F6 < 300 ;
F1 + F5 < 5 ;
F1 + F5 + F7 < 300 ;
F3 < 500 ;
F2 + F6 < 200 ;
F1 + F5 + F7 < 300 ;
F2 + F3 + F4 + F6 < 600 ;
F2 + F3 + F4 + F6 < 350 ;
50 * X1 + 50 * X2 + 50 * X3 + 50 * X4 + 50 * X5 + 50 * X6 +
50 * X7 + 50 * X8 + 50 * X9 + 50 * X10 + 50 * X11 +
50 * X12 + 50 * X13 + 50 * X14 + 50 * X15 + 50 * X16 +
50 * X17 + 50 * X18  < 1000 ;
END
SLB X1 0
SLB X2 0
SLB X3 0
SLB X4 0
SLB X5 0
SLB X6 0
SLB X7 0
SLB X8 0
SLB X9 0
SLB X10 0
SLB X11 0
SLB X12 0
SLB X13 0
SLB X14 0
SLB X15 0
SLB X16 0
SLB X17 0
SLB X18 0
SLB F1 0
SLB F2 0
SLB F3 0
SLB F4 0
SLB F5 0
SLB F6 0
SLB F7 0
LEAVE
```

```
 SOLUTION STATUS:  OPTIMAL TO TOLERANCES.  DUAL CONDITIONS:  SATISFIED.

             OBJECTIVE FUNCTION VALUE

        1)       155.087321

   VARIABLE        VALUE          REDUCED COST
        R1          0.000000          0.000000
```

F1	0.000000	0.000000
R2	0.000000	0.000000
F2	0.000000	0.000000
R3	0.173782	0.000000
F3	150.000000	0.000000
R4	0.000000	0.000000
F4	0.000000	0.000000
R5	0.000000	0.000000
F5	0.000000	0.000000
R6	0.645100	0.000000
F6	200.000000	0.000000
R7	1.000000	0.000000
F7	300.000000	0.000000
X1	0.000000	0.000000
X2	0.000000	0.000000
X3	0.000000	0.000000
X7	0.000000	0.000000
X8	0.000000	0.000000
X9	1.174248	0.000049
X10	0.000000	0.000000
X11	5.000000	0.000000
X14	4.000000	0.000000
X17	0.000000	0.000000
X18	0.000000	0.000000
X13	5.174223	0.000000
X15	1.825779	0.000045
X16	2.825751	0.000000
X12	0.000000	0.000000
X4	0.000000	0.000000
X5	0.000000	0.000000
X6	0.000000	0.000000

ROW	SLACK OR SURPLUS	PRICE
2)	0.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	150.000000
5)	0.000000	0.000000
6)	0.000000	0.000000
7)	0.000000	200.000000
8)	0.000000	300.000000
9)	0.000000	297.249685
10)	0.000000	297.249685
11)	0.000000	0.000000
12)	1.000000	0.000000
13)	0.000000	0.000000
14)	0.000000	142.162369
15)	0.700000	0.000000
16)	1.000000	0.000000
17)	0.182575	0.000000
18)	1.000000	0.000000
19)	0.000000	142.162369

20)	0.400000	0.000000
21)	0.182578	0.000000
22)	0.000000	142.162369
23)	0.017422	0.000000
24)	0.017425	0.000000
25)	0.000000	297.249685
26)	0.000000	297.249685
27)	0.000000	0.000000
28)	0.000000	0.000000
29)	0.000000	-13.177030
30)	0.000000	-15.783763
31)	0.000000	-2.881763
32)	0.000000	0.000000
33)	0.000000	-7.094653
34)	0.000000	-15.783763
35)	1.174248	0.000000
36)	0.000000	-15.783763
37)	5.000000	0.000000
38)	0.000000	-11.439208
39)	5.174223	0.000000
40)	4.000000	0.000000
41)	1.825779	0.000000
42)	2.825751	0.000000
43)	0.000000	0.000000
44)	0.000000	0.000000
45)	350.000000	0.000000
46)	350.000000	0.000000
47)	5.000000	0.000000
48)	100.000000	0.000000
49)	300.000000	0.000000
50)	350.000000	0.000000
51)	5.000000	0.000000
52)	100.000000	0.000000
53)	5.000000	0.000000
54)	0.000000	0.000000
55)	350.000000	0.000000
56)	0.000000	0.471318
57)	0.000000	1.000000
58)	250.000000	0.000000
59)	0.000000	0.173782
60)	0.000000	-1.000000
61)	0.000000	-0.645100
62)	150.000000	0.000000
63)	0.000000	-0.173782
64)	0.000000	-1.000000
65)	200.000000	0.000000
66)	300.000000	0.000000
67)	0.000000	0.315675

A.4.7 Output From Maxflo

```
   Original Network

NORMAL STATISTICS
 Mean:   145.455
 Std. Dev.:  181.803
 Confidence intvl. (+-):  3.56333

 Reliability:
 0.435300

     Network After Investment Strategy 1

NORMAL STATISTICS
 Mean:   148.476
 Std. Dev.:  185.387
 Confidence intvl. (+-):  3.63358

 Reliability:
 0.435300

     Network After Investment Strategy 3

NORMAL STATISTICS
 Mean:   466.645
 Std. Dev.:  120.940
 Confidence intvl. (+-):  2.37042

 Reliability:
 1.00000
```

A.5 Calculating Lower Bound Using Netflo

A.5.1 Network 2

```
title 'Network 2';
title3 'Nodes for Network 2';
   data noded;
      input _node_ $ _sd_;
      cards;
   s 0
   t 0
;

title3 'Arcs for Network 2';
   data arcd1;
      input  _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   s n1 0 . . xs1
   n1 n2 0 . . x12
   n1 n3 0 . . x13
   n2 n4 0 400 . x24
   n3 n5 0 300 . x35
   n4 n6 0 500 . x46
   n5 n6 0 300 . x56
   n6 t 0 . . x6t
   t s -1 . . xts
   . . 0 400 . f1
   . . 0 300 . f2
;

title3 'Side constraints';
data cond1;
   input f1 f2 x6t _type_ $ _rhs_;
   cards;
0.2 0.42 -1 EQ 0
;

proc netflow
  nodedata=noded
  arcdata=arcd1
  condata=cond1
  conout=solution
  maxflow
  sourcenode=s
  sinknode=t
  namectrl=1;

print problem;

proc print data=solution;
title3 'lower bound';
```

```
NOTE: Number of arcs= 9 .
NOTE: Number of nonarc variables= 2 .
NOTE: Number of iterations performed (neglecting any constraints)= 9 .
NOTE: Of these, 6 were degenerate.
NOTE: Maximal flow= 700 .
NOTE: Optimum (neglecting any constraints) found.
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 0 .
NOTE: Number of == side constraints= 1 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 3 .
NOTE: Number of iterations, optimizing with constraints= 5 .
NOTE: Of these, 0 were degenerate.
NOTE: Maximal flow= 206 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 0 .
NOTE: The data set WORK.SOLUTION has 11 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed page 1.

46        proc print data=solution;
47        title3 'lower bound';
NOTE: The PROCEDURE PRINT printed page 2.

NOTE: SAS Institute Inc., SAS Circle, PO Box 8000, Cary, NC 27512-8000
```

A.5.2 Network 3

```
title 'Network 3';
title3 'Nodes for Network 3';
   data noded;
      input _node_ $ _sd_;
      cards;
   s 0
   t 0
;

title3 'Arcs for Network 3';
   data arcd1;
      input  _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   s n1 0 5 . xs1
   s n2 0 6 . xs2
   n1 n2 0 5 . x12
   n1 n3 0 2 . x13
   n1 n4 0 4 . x14
   n2 n4 0 5 . x24
   n3 t 0 4 . x3t
   n4 n3 0 5 . x43
   n4 t 0 7 . x4t
   t s -1 . . xts
   . . . . . f1
   . . . . . f2
   . . . . . f3
   . . . . . f4
   . . . . . f5
   . . . . . f6
   . . . . . f7
;

title3 'Side constraints';
data cond1;
   input f1 f2 f3 f4 f5 f6 f7 x3t x4t _type_ $ _rhs_;
   cards;
1 1 1 1 1 . . . . LE 5
. . . . . 1 1 . . LE 6
1 1 . . . . . . . LE 5
. . 1 . . . . . . LE 2
. . . 1 1 . . . . LE 4
1 1 . . . 1 1 . . LE 5
1 . . 1 . 1 . . . LE 5
1 . 1 1 . 1 . . . LE 4
. 1 . . 1 . 1 . . LE 7
. . 0.9 . 1 . 0.05 -1 -1 EQ 0
;

proc netflow
```

```
  nodedata=noded
  arcdata=arcd1
  condata=cond1
  conout=solution
  maxflow
  sourcenode=s
  sinknode=t
  namectrl=1;

print problem;

proc print data=solution;
sum _fcost_;
title3 'lower bound ';


NOTE: Number of nodes= 6 .
NOTE: Number of arcs= 10 .
NOTE: Number of nonarc variables= 7 .
NOTE: Number of iterations performed (neglecting any constraints)= 9 .
NOTE: Of these, 4 were degenerate.
NOTE: Maximal flow= 10 .
NOTE: Optimum (neglecting any constraints) found.
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 9 .
NOTE: Number of == side constraints= 1 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 31 .
NOTE: Number of iterations, optimizing with constraints= 7 .
NOTE: Of these, 1 were degenerate.
NOTE: Maximal flow= 5.05 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 0 .
NOTE: The data set WORK.SOLUTION has 17 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed pages 1-2.
```

A.5.3 Network 4

```
title 'Network 4';
title3 'Nodes for Network 4';
   data noded;
      input _node_ $ _sd_;
      cards;
   n1 0
   n6 0
;

title3 'Arcs for Network 4';
   data arcd1;
      input  _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   n1 n2a 0 500 . x12a
   n1 n3a 0 300 . x13a
   n1 n4a 0 600 . x14a
   n1 n5a 0 5 . x15a
   n2a n2b 0 500 . x2a2b
   n2a n3a 0 5 . x2a3a
   n2b n5a 0 500 . x2b5a
   n3a n3b 0 300 . x3a3b
   n3a n4a 0 5 . x3a4a
   n3b n5a 0 200 . x3b5a
   n4a n4b 0 300 . x4a4b
   n4b n6 0 300 . x4b6
   n5a n5b 0 600 . x5a5b
   n5b n6 0 350 . x5b6
   n6 n1 -1 . . x61
   . . . . . f1
   . . . . . f2
   . . . . . f3
   . . . . . f4
   . . . . . f5
   . . . . . f6
   . . . . . f7
;

title3 'Side constraints';
data cond1;
   input
f1 f2 f3 f4 f5 f6 f7 x4b6 x5b6 _type_ $ _rhs_;
   cards;
1 1 1 1 1 1 1 . . LE 1000
1 1 1 . . . . . . LE 500
. . . 1 . . . . . LE 5
. . . . 1 1 . . . LE 300
. . . . . . 1 . . LE 600
1 1 . . . . . . . LE 5
1 1 . . 1 1 . . . LE 300
```

```
1 . . . 1 . . . . LE 5
1 . . . 1 . 1 . . LE 300
. . 1 . . . . . . LE 500
. 1 . . . 1 . . . LE 200
. 1 1 1 . 1 . . . LE 350
. . 0.1008 . . 0.1176 0.3 -1 -1 EQ 0
;

proc netflow
  nodedata=noded
  arcdata=arcd1
  condata=cond1
  conout=solution
  maxflow
  sourcenode=n1
  sinknode=n6
  namectrl=1;

print problem;

proc print data=solution;
sum _fcost_;
title3 'lower bound ';


NOTE: Number of nodes= 10 .
NOTE: Number of arcs= 15 .
NOTE: Number of nonarc variables= 7 .
NOTE: Number of iterations performed (neglecting any constraints)= 12 .
NOTE: Of these, 8 were degenerate.
NOTE: Maximal flow= 650 .
NOTE: Optimum (neglecting any constraints) found.
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 12 .
NOTE: Number of == side constraints= 1 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 37 .
NOTE: Number of iterations, optimizing with constraints= 6 .
NOTE: Of these, 0 were degenerate.
NOTE: Maximal flow= 128.64 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 0 .
NOTE: The data set WORK.SOLUTION has 22 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed pages 1-3.
```

A.6 *Calculating Upper Bound Using Microsolv*

A.6.1 *Network 2* Network 2, Page 1

PRIMAL SIMPLEX ALGORITHM

OPTIMUM SOLUTION OBTAINED.

ARC PARAMETERS AND FLOWS
SOLUTION COST = -380

ARCS THAT START AT s

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n1	1	0	999999	0	1	380

ARCS THAT START AT n1

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n2	2	0	999999	0	1	200
n3	3	0	999999	0	1	180

ARCS THAT START AT n2

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n4	4	0	200	0	1	200

ARCS THAT START AT n3

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n5	5	0	210	0	1	180

ARCS THAT START AT n4

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n6	6	0	200	0	1	200

ARCS THAT START AT n5

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n6	7	0	180	0	1	180

ARCS THAT START AT n6

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
t	8	0	999999	0	1	380

ARCS THAT START AT t

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
s	9	0	999999	-1	1	380

ARCS THAT START AT SLACK

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
s	10	0	0	99999	1	0
n1	11	0	0	99999	1	0
n2	12	0	0	99999	1	0
n3	13	0	0	99999	1	0
n4	14	0	0	99999	1	0

Network 2, Page 2

n5	15	0	0	99999	1	0
n6	16	0	0	99999	1	0
t	17	0	0	99999	1	0

NODE PARAMETERS

NODE	NAME	POTENTIAL	EXT.FLOW
1	s	99999	0
2	n1	99999	0
3	n2	99999	0
4	n3	99999	0
5	n4	100000	0
6	n5	99999	0
7	n6	100000	0
8	t	100000	0
9	SLACK	0	0

A.6.2 Network 3 Network 3, Page 1

PRIMAL SIMPLEX ALGORITHM

OPTIMUM SOLUTION OBTAINED.

ARC PARAMETERS AND FLOWS
SOLUTION COST = -5.6

ARCS THAT START AT s

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n1	1	0	5	0	1	5
n2	2	0	.6	0	1	.6

ARCS THAT START AT n1

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n3	3	0	2	0	1	1
n4	4	0	4	[illegible]	1	4

ARCS THAT START AT n2

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n4	5	0	2.5	0	1	.6

ARCS THAT START AT n3

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
t	6	0	3.6	0	1	1

ARCS.THAT START AT n4

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
t	7	0	7	0	1	4.6

ARCS THAT START AT t

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
s	8	0	999999	-1	1	5.6

ARCS THAT START AT SLACK

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
s	9	0	0	99999	1	0
n1	10	0	0	99999	1	0
n2	11	0	0	99999	1	0
n3	12	0	0	99999	1	0
n4	13	0	0	99999	1	0
t	14	0	0	99999	1	0

A.6.3 *Network 4* Network 4, Page 1

PRIMAL SIMPLEX ALGORITHM

OPTIMUM SOLUTION OBTAINED.

ARC PARAMETERS AND FLOWS
SOLUTION COST = -360

ARCS THAT START AT n1a

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n1b	1	0	1000	0	1	360

ARCS THAT START AT n2a

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n2b	2	0	150	0	1	150

ARCS THAT START AT n2b

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n5a	3	0	300	0	1	150
n3a	4	0	0	0	1	0

ARCS THAT START AT n3a

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n3b	5	0	210	0	1	60

ARCS THAT START AT n3b

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n4a	6	0	0	0	1	0
n5a	7	0	60	0	1	60

ARCS THAT START AT n4a

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n4b	8	0	150	0	1	150

ARCS THAT START AT n4b

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n6a	9	0	180	0	1	150

ARCS THAT START AT n5a

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n5b	10	0	480	0	1	210

ARCS THAT START AT n5b

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n6a	11	0	245	0	1	210

ARCS THAT START AT n6a

Network 4, Page 2

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n6b	12	0	999999	0	1	360

ARCS THAT START AT n1b

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n2a	13	0	500	0	1	150
n3a	14	0	300	0	1	60
n4a	15	0	600	0	1	150
n5a	16	0	0	0	1	0

ARCS THAT START AT n6b

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n1a	17	0	999999	-1	1	360

ARCS THAT START AT SLACK

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
n1a	18	0	0	99999	1	0
n2a	19	0	0	99999	1	0
n2b	20	0	0	99999	1	0
n3a	21	0	0	99999	1	0
n3b	22	0	0	99999	1	0
n4a	23	0	0	99999	1	0
n4b	24	0	0	99999	1	0
n5a	25	0	0	99999	1	0
n5b	26	0	0	99999	1	0
n6a	27	0	0	99999	1	0
n1b	28	0	0	99999	1	0
n6b	29	0	0	99999	1	0

A.7 Investment Strategy Model 1 Using Netflo

A.7.1 Network 2

```
title 'Network 2';
title3 'Nodes for Network 2';
   data noded;
      input _node_ $ _sd_;
      cards;
   s 0
   t 0
;

title3 'Arcs for Network 2';
   data arcd1;
      input  _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   s n1 0 . . xs1
   n1 n2 0 . . x12
   n1 n3 0 . . x13
   n2 n4 0 400 . x24
   n3 n5 0 300 . x35
   n4 n6 0 500 . x46
   n5 n6 0 300 . x56
   n6 t 0 . . x6t
   t s -1 . . xts
   . . 0 . . f1
   . . 0 . . f2
   . . 0 20 . d24
   . . 0 20 . d35
   . . 0 20 . d46
   . . 0 20 . d56
;

title3 'Side constraints';
data cond1;
   input f1 f2 d24 d35 d46 d56 x6t _type_ $ _rhs_;
   cards;
1 . -1 . . . . LE 400
1 . . . -1 . . LE 500
. 1 . -1 . . . LE 300
. 1 . . . -1 . LE 300
. . 50 50 50 50 . EQ 1000
0.2 0.42 . . . . -1 EQ 0
;

proc netflow
  nodedata=noded
  arcdata=arcd1
  condata=cond1
  conout=solution
  maxflow
  sourcenode=s
  sinknode=t
  namectrl=1;

print problem;

proc print data=solution;
title3 'lower bound after increasing capacity ';

NOTE: Number of arcs= 9 .
NOTE: Number of nonarc variables= 6 .
NOTE: Number of iterations performed (neglecting any constraints)= 9 .
```

```
NOTE: Of these, 6 were degenerate.
NOTE: Maximal flow= 700 .
NOTE: Optimum (neglecting any constraints) found.
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 4 .
NOTE: Number of == side constraints= 2 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 15 .
NOTE: Number of iterations, optimizing with constraints= 10 .
NOTE: Of these, 3 were degenerate.
NOTE: Maximal flow= 210.2 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 0 .
NOTE: The data set WORK.SOLUTION has 15 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed pages 1-2.
```

lower bound after increasing capacity

OBS	_FROM_	_TO_	_COST_	_CAPAC_	_LO_	_NAME_	_SUPPLY_	_DEMAND_	_FLOW_	_STATUS_
1	s	n1	0	99999999	0	XS1	99999998	.	210.2	KEY_ARC BASIC
2	n1	n2	0	99999999	0	X12	.	.	210.2	KEY_ARC BASIC
3	n1	n3	0	99999999	0	X13	.	.	0.0	KEY_ARC BASIC
4	n2	n4	0	400	0	X24	.	.	210.2	NONKEY ARC BASIC
5	n3	n5	0	300	0	X35	.	.	0.0	KEY_ARC BASIC
6	n4	n6	0	500	0	X46	.	.	210.2	KEY_ARC BASIC
7	n5	n6	0	300	0	X56	.	.	0.0	LOWERBD NONBASIC
8	t	s	-1	99999999	0	XTS	.	.	0.0	LOWERBD NONBASIC
9	n6	t	0	99999999	0	X6T	.	99999998	210.2	KEY_ARC BASIC
10			0	20	0	D24	.	.	0.0	LOWERBD NONBASIC
11			0	20	0	D35	.	.	10.0	NONKEY BASIC
12			0	20	0	D46	.	.	0.0	LOWERBD NONBASIC
13			0	20	0	D56	.	.	10.0	NONKEY BASIC
14			0	99999999	0	F1	.	.	400.0	NONKEY BASIC
15			0	99999999	0	F2	.	.	310.0	NONKEY BASIC

A.7.2 Network 3

```
title 'Network 3';
title3 'Nodes for Network 3';
   data noded;
      input _node_ $ _sd_;
      cards;
   s 0
   t 0
;

title3 'Arcs for Network 3';
   data arcd1;
      input  _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   s n1 0 . . xs1
   s n2 0 . . xs2
   n1 n2 0 . . x12
   n1 n3 0 . . x13
   n1 n4 0 . . x14
   n2 n4 0 . . x24
   n3 t 0 . . x3t
   n4 n3 0 . . x43
   n4 t 0 . . x4t
   . . 0 . . f1
   . . 0 . . f2
   . . 0 . . f3
   . . 0 . . f4
   . . 0 . . f5
   . . 0 . . f6
   . . 0 . . f7
   . . 0 20 . ds1
   . . 0 20 . ds2
   . . 0 20 . d12
   . . 0 20 . d13
   . . 0 20 . d14
   . . 0 20 . d24
   . . 0 20 . d3t
   . . 0 20 . d43
   . . 0 20 . d4t
;

title3 'Side constraints';
data cond1;
   input
f1 f2 f3 f4 f5 f6 f7 ds1 ds2 d12 d13 d14 d24 d3t d43 d4t x3t x4t _type_ $ _rhs_;
   cards;
1 1 1 1 1 . . -1 . . . . . . . . . . LE 5
. . . . . 1 1 . -1 . . . . . . . . . LE 6
1 1 . . . . . . . -1 . . . . . . . . LE 5
. . 1 . . . . . . . -1 . . . . . . . LE 2
. . . 1 1 . . . . . . -1 . . . . . . LE 4
1 1 . . . 1 1 . . . . . -1 . . . . . LE 5
1 . . 1 . 1 . . . . . . . -1 . . . . LE 5
1 . 1 1 . 1 . . . . . . . . -1 . . . LE 4
. 1 . . 1 . 1 . . . . . . . . -1 . . LE 7
. . . . . . . 5 5 5 5 5 5 5 5 5 . . EQ 100
. . 0.9 . 1 . 0.05 . . . . . . . . . -1 -1 EQ 0
;

proc netflow
  nodedata=noded
  arcdata=arcd1
  condata=cond1
```

```
  conout=solution
  maxflow
  sourcenode=s
  sinknode=t
  namectrl=1;

print problem;

proc print data=solution;
sum _fcost_;
title3 'lower bound after increasing capacity';

NOTE: Number of nodes= 6 .
NOTE: Number of arcs= 9 .
WARNING: The network has no arcs that have non-INFINITY capacities. MAXFLOW was specified. The maximal flow = INFINITY.
NOTE: Number of nonarc variables= 16 .
NOTE: Number of iterations performed (neglecting any constraints)= 6 .
NOTE: Of these, 4 were degenerate.
NOTE: Maximal flow= 99999998 .
NOTE: Optimum (neglecting any constraints) found.
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 9 .
NOTE: Number of == side constraints= 2 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 49 .
NOTE: Number of iterations, optimizing with constraints= 8 .
NOTE: Of these, 0 were degenerate.
NOTE: Maximal flow= 13.6 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 99999984.4 .
NOTE: The data set WORK.SOLUTION has 25 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed pages 1-3.
```

lower bound after increasing capacity

OBS	_FROM_	_TO_	_COST_	_CAPAC_	_LO_	_NAME_	_SUPPLY_	_DEMAND_	_FLOW_	_STATUS_
1	s	n1	0	99999999	0	XS1	99999998	.	13.6	KEY_ARC BASIC
2	s	n2	0	99999999	0	XS2	99999998	.	0.0	KEY_ARC BASIC
3	n1	n2	0	99999999	0	X12	.	.	0.0	LOWERBD NONBASIC
4	n1	n3	0	99999999	0	X13	.	.	13.6	KEY_ARC BASIC
5	n4	n3	0	99999999	0	X43	.	.	0.0	LOWERBD NONBASIC
6	n1	n4	0	99999999	0	X14	.	.	0.0	KEY_ARC BASIC
7	n2	n4	0	99999999	0	X24	.	.	0.0	LOWERBD NONBASIC
8	n3	t	0	99999999	0	X3T	.	99999998	13.6	KEY_ARC BASIC
9	n4	t	0	99999999	0	X4T	.	9999999	0.0	LOWERBD NONBASIC
10			0	20	0	D12	.	.	0.0	LOWERBD NONBASIC
11			0	20	0	D13	.	.	2.0	NONKEY BASIC
12			0	20	0	D14	.	.	6.0	NONKEY BASIC
13			0	20	0	D24	.	.	0.0	LOWERBD NONBASIC
14			0	20	0	D3T	.	.	0.0	LOWERBD NONBASIC
15			0	20	0	D43	.	.	0.0	LOWERBD NONBASIC
16			0	20	0	D4T	.	.	3.0	NONKEY BASIC
17			0	20	0	DS1	.	.	9.0	NONKEY BASIC
18			0	20	0	DS2	.	.	0.0	LOWERBD NONBASIC
19			0	99999999	0	F1	.	.	0.0	LOWERBD NONBASIC
20			0	99999999	0	F2	.	.	0.0	LOWERBD NONBASIC
21			0	99999999	0	F3	.	.	4.0	NONKEY BASIC
22			0	99999999	0	F4	.		0.0	LOWERBD NONBASIC
23			0	99999999	0	F5	.	.	10.0	NONKEY BASIC
24			0	99999999	0	F6	.	.	0.0	LOWERBD NONBASIC
25			0	99999999	0	F7	.	.	0.0	LOWERBD NONBASIC

A.7.3 Network 4

```
title 'Network 4';
title3 'Nodes for Network 4';
   data noded;
      input _node_ $ _sd_;
      cards;
   n1a 0
   n6 0
;

title3 'Arcs for Network 4';
   data arcd1;
      input  _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   n1a n1b 0 . . x1a1b
   n1b n2a 0 . . x1b2a
   n1b n3a 0 . . x1b3a
   n1b n4a 0 . . x1b4a
   n1b n5a 0 . . x1b5a
   n2a n2b 0 . . x2a2b
   n2a n3a 0 . . x2a3a
   n2b n5a 0 . . x2b5a
   n3a n3b 0 . . x3a3b
   n3a n4a 0 . . x3a4a
   n3b n5a 0 . . x3b5a
   n4a n4b 0 . . x4a4b
   n4b n6 0 . . x4b6
   n5a n5b 0 . . x5a5b
   n5b n6 0 . . x5b6
   . . . . . f1
   . . . . . f2
   . . . . . f3
   . . . . . f4
   . . . . . f5
   . . . . . f6
   . . . . . f7
   . . . . . d1a1b
   . . . . . d1b2a
   . . . . . d1b3a
   . . . . . d1b4a
   . . . . . d1b5a
   . . . . . d2a2b
   . . . . . d2b3a
   . . . . . d2b5a
   . . . . . d3a3b
   . . . . . d3b4a
   . . . . . d3b5a
   . . . . . d4a4b
   . . . . . d4b6
   . . . . . d5a5b
   . . . . . d5b6
;

title3 'Side constraints';
data cond1;
   input
f1 f2 f3 f4 f5 f6 f7
d1a1b d1b2a d1b3a d1b4a d1b5a d2a2b d2b3a d2b5a d3a3b d3b4a d3b5a d4a4b
d4b6 d5a5b d5b6 x4b6 x5b6 _type_ $ _rhs_;
   cards;
1 1 1 1 1 1 1 -1 . . . . . . . . . . . . . . . . LE 1000
1 1 1 . . . . . -1 . . . . . . . . . . . . . . . LE 500
. . . 1 . . . . . . . -1 . . . . . . . . . . . . LE 5
```

```
. . . . 1 1 . . . -1 . . . . . . . . . . . . . . . LE 300
. . . . . . 1 . . . -1 . . . . . . . . . . . . . . LE 600
1 1 1 . . . . . . . . . -1 . . . . . . . . . . . . LE 500
1 1 . . . . . . . . . . . . -1 . . . . . . . . . . LE 5
1 1 . . 1 1 . . . . . . . . . . -1 . . . . . . . . LE 300
1 . . . 1 . . . . . . . . . . . . -1 . . . . . . . LE 5
1 . . . 1 . 1 . . . . . . . . . . . . -1 . . . . . LE 300
. . 1 . . . . . . . . . . . . -1 . . . . . . . . . LE 500
. 1 . . . 1 . . . . . . . . . . . . -1 . . . . . . LE 200
1 . . . 1 . . . . . . . . . . . . . . . -1 . . . . LE 300
. 1 1 1 . 1 . . . . . . . . . . . . . . . -1 . . . LE 600
. 1 1 1 . 1 . . . . . . . . . . . . . . . . -1 . . LE 350
. . . . . . . 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 . . LE 1000
. . 0.1008 . . 0.1176 0.3 . . . . . . . . . . . . . . . . -1 -1 EQ 0
;

proc netflow
  nodedata=noded
  arcdata=arcd1
  condata=cond1
  conout=solution
  maxflow
  sourcenode=n1a
  sinknode=n6
  namectrl=1;

print problem;

proc print data=solution;
sum _fcost_;
title3 'lower bound after increasing capacity';

NOTE: Number of nonarc variables= 22 .
NOTE: Number of iterations performed (neglecting any constraints)= 11 .
NOTE: Of these, 9 were degenerate.
NOTE: Maximal flow= 99999998 .
NOTE: Optimum (neglecting any constraints) found.
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 16 .
NOTE: Number of == side constraints= 1 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 77 .
NOTE: Number of iterations, optimizing with constraints= 8 .
NOTE: Of these, 2 were degenerate.
NOTE: Maximal flow= 131.64 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 0 .
NOTE: The data set WORK.SOLUTION has 37 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed pages 1-5.
```

lower bound after increasing capacity

OBS	_FROM_	_TO_	_COST_	_CAPAC_	_LO_	_NAME_	_SUPPLY_	_DEMAND_	_FLOW_	_STATUS_	
1	n1a	n1b	0	99999999	0	X1A1B	99999998	.	131.64	KEY_ARC	BASIC
2	n1b	n2a	0	99999999	0	X1B2A	.	.	0.00	KEY_ARC	BASIC
3	n2a	n2b	0	99999999	0	X2A2B	.	.	0.00	KEY_ARC	BASIC
4	n1b	n3a	0	99999999	0	X1B3A	.	.	0.00	KEY_ARC	BASIC
5	n2a	n3a	0	99999999	0	X2A3A	.	.	0.00	LOWERBD	NONBASIC
6	n3a	n3b	0	99999999	0	X3A3B	.	.	0.00	KEY_ARC	BASIC
7	n1b	n4a	0	99999999	0	X1B4A	.	.	131.64	KEY_ARC	BASIC
8	n3a	n4a	0	99999999	0	X3A4A	.	.	0.00	LOWERBD	NONBASIC
9	n4a	n4b	0	99999999	0	X4A4B	.	.	131.64	KEY_ARC	BASIC
10	n1b	n5a	0	99999999	0	X1B5A	.	.	0.00	KEY_ARC	BASIC
11	n2b	n5a	0	99999999	0	X2B5A	.	.	0.00	LOWERBD	NONBASIC

12	n3b	n5a	0	99999999	0	X3B5A	.	.	0.00	LOWERBD	NONBASIC
13	n5a	n5b	0	99999999	0	X5A5B	.	.	0.00	KEY_ARC	BASIC
14	n4b	n6	0	99999999	0	X4B6	.	99999998	131.64	NONKEY ARC	BASIC
15	n5b	n6	0	99999999	0	X5B6	.	99999998	0.00	LOWERBD	NONBASIC
16			0	99999999	0	D1A1B	.	.	0.00	LOWERBD	NONBASIC
17			0	99999999	0	D1B2A	.	.	0.00	LOWERBD	NONBASIC
18			0	99999999	0	D1B3A	.		0.00	LOWERBD	NONBASIC
19			0	99999999	0	D1B4A	.	.	0.00	LOWERBD	NONBASIC
20			0	99999999	0	D1B5A	.	.	0.00	LOWERBD	NONBASIC
21			0	99999999	0	D2A2B	.	.	0.00	LOWERBD	NONBASIC
22			0	99999999	0	D2B3A	.	.	0.00	LOWERBD	NONBASIC
23			0	99999999	0	D2B5A	.	.	0.00	LOWERBD	NONBASIC
24			0	99999999	0	D3A3B	.	.	0.00	LOWERBD	NONBASIC
25			0	99999999	0	D3B4A	.	.	0.00	LOWERBD	NONBASIC
26			0	99999999	0	D3B5A	.	.	0.00	LOWERBD	NONBASIC
27			0	99999999	0	D4A4B	.	.	10.00	NONKEY	BASIC
28			0	99999999	0	D4B6	.	.	10.00	NONKEY	BASIC
29			0	99999999	0	D5A5B	.	.	0.00	LOWERBD	NONBASIC
30			0	99999999	0	D5B6	.	.	0.00	LOWERBD	NONBASIC
31			0	99999999	0	F1	.	.	0.00	LOWERBD	NONBASIC
32			0	99999999	0	F2	.	.	0.00	LOWERBD	NONBASIC
33			0	99999999	0	F3	.	.	150.00	NONKEY	BASIC
34			0	99999999	0	F4	.	.	0.00	LOWERBD	NONBASIC
35			0	99999999	0	F5	.	.	0.00	LOWERBD	NONBASIC
36			0	99999999	0	F6	.	.	200.00	NONKEY	BASIC
37			0	99999999	0	F7	.	.	310.00	NONKEY	BASIC

A.8 Investment Strategy Model 4 Using Netflo

A.8.1 Network 2

```
title 'Network 2';
title3 'Nodes for Network 2';
   data noded;
      input _node_ $ _sd_;
      cards;
   s 0
   t 0
;

title3 'Arcs for Network 2';
   data arcd1;
      input  _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   s n1 0 . . xs1
   n1 n2 0 . . x12
   n1 n3 0 . . x13
   n2 n4 0 400 . x24
   n3 n5 0 300 . x35
   n4 n6 0 500 . x46
   n5 n6 0 300 . x56
   n6 t 0 . . x6t
   t s -1 . . xts
   . . 0 5 0 G24
   . . 0 3 0 G35
   . . 0 6 0 G46
   . . 0 4 0 G56
;

title3 'Side constraints';
data cond1;
   input xs1 x12 x13 x24 x35 x46 x56 x6t g24 g35 g46 g56 _type_ $ _rhs_;
   cards;
. . . 1 . . . . -40 . . . LE 200
. . . . 1 . . . . -30 . . LE 210
. . . . . 1 . . . . -50 . LE 200
. . . . . . 1 . . . . -30 LE 180
. . . . . . . . 5 5 5 5 LE 100
;

proc netflow
  nodedata=noded
  arcdata=arcd1
  condata=cond1
  conout=solution
  maxflow
  sourcenode=s
  sinknode=t
  namectrl=1;

print problem;

proc print data=solution;
title3 'upper bound after improving reliability';

NOTE: Number of arcs= 9 .
NOTE: Number of nonarc variables= 4 .
NOTE: Number of iterations performed (neglecting any constraints)= 9 .
NOTE: Of these, 6 were degenerate.
NOTE: Maximal flow= 700 .
NOTE: Optimum (neglecting any constraints) found.
```

```
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 5 .
NOTE: Number of == side constraints= 0 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 12 .
NOTE: Number of iterations, optimizing with constraints= 7 .
NOTE: Of these, 1 were degenerate.
NOTE: Maximal flow= 700 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 0 .
NOTE: The data set WORK.SOLUTION has 13 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed pages 1-2.
```

upper bound after improving reliability

OBS	_FROM_	_TO_	_COST_	_CAPAC_	_LO_	_NAME_	_SUPPLY_	_DEMAND_	_FLOW_	_STATUS_
1	s	n1	0	99999999	0	XS1	99999998	.	700	KEY_ARC BASIC
2	n1	n2	0	99999999	0	X12	.	.	400	KEY_ARC BASIC
3	n1	n3	0	99999999	0	X13	.	.	300	KEY_ARC BASIC
4	n2	n4	0	400	0	X24	.	.	400	UPPERBD NONBASIC
5	n3	n5	0	300	0	X35	.	.	300	KEY_ARC BASIC
6	n4	n6	0	500	0	X46	.	.	400	KEY_ARC BASIC
7	n5	n6	0	300	0	X56	.	.	300	NONKEY ARC BASIC
8	t	s	-1	99999999	0	XTS	.	.	0	LOWERBD NONBASIC
9	n6	t	0	99999999	0	X6T	.	99999998	700	KEY_ARC BASIC
10			0	5	0	G24	.	.	5	NONKEY BASIC
11			0	3	0	G35	.	.	3	UPPERBD NONBASIC
12			0	6	0	G46	.	.	4	NONKEY BASIC
13			0	4	0	G56	.	.	4	NONKEY BASIC

A.8.2 Network 3

```
title 'Network 3';
title3 'Nodes for Network 3';
   data noded;
      input _node_ $ _sd_;
      cards;
   s 0
   t 0
;

title3 'Arcs for Network 3';
   data arcd1;
      input  _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   s n1 0 5 . xs1
   s n2 0 6 . xs2
   n1 n2 0 5 . x12
   n1 n3 0 2 . x13
   n1 n4 0 4 . x14
   n2 n4 0 5 . x24
   n3 t 0 4 . x3t
   n4 n3 0 5 . x43
   n4 t 0 7 . x4t
   t s -1 . . xts
   . . 0 9 . gs2
   . . 0 10 . g12
   . . 0 5 . g24
   . . 0 1 . g3t
   . . 0 10 . g43
;

title3 'Side constraints';
data cond1;
   input xs2 x12 x24 x3t x43 gs2 g12 g24 g3t g43 _type_ $ _rhs_;
   cards;
1 . . . . -0.6 . . . . LE 0.6
. 1 . . . . -0.5 . . . LE 0
. . 1 . . . . -0.5 . . LE 2.5
. . . 1 . . . . -0.4 . LE 3.6
. . . . 1 . . . . -0.5 LE 0
. . . . . 5 5 5 5 5 LE 100
;

proc netflow
  nodedata=noded
  arcdata=arcd1
  condata=cond1
  conout=solution
  maxflow
  sourcenode=s
  sinknode=t
  namectrl=1;

print problem;

proc print data=solution;
sum _fcost_;
title3 'upper bound after improving reliability ';

NOTE: Number of arcs= 10 .
NOTE: Number of nonarc variables= 5 .
NOTE: Number of iterations performed (neglecting any constraints)= 9 .
NOTE: Of these, 4 were degenerate.
```

```
NOTE: Maximal flow= 10 .
NOTE: Optimum (neglecting any constraints) found.
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 6 .
NOTE: Number of == side constraints= 0 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 15 .
NOTE: Number of iterations, optimizing with constraints= 8 .
NOTE: Of these, 2 were degenerate.
NOTE: Maximal flow= 10 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 0 .
NOTE: The data set WORK.SOLUTION has 15 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed pages 1-2.
```

'upper bound after improving reliability

OBS	_FROM_	_TO_	_COST_	_CAPAC_	_LO_	_NAME_	_SUPPLY_	_DEMAND_	_FLOW_	_STATUS_
1	s	n1	0	5	0	XS1	99999998	.	5.00000	UPPERBD NONBASIC
2	s	n2	0	6	0	XS2	99999998	.	5.00000	KEY_ARC BASIC
3	n1	n2	0	5	0	X12	.	.	0.00000	LOWERBD NONBASIC
4	n1	n3	0	2	0	X13	.	.	2.00000	UPPERBD NONBASIC
5	n4	n3	0	5	0	X43	.	.	1.60000	KEY_ARC BASIC
6	n1	n4	0	4	0	X14	.	.	3.00000	KEY_ARC BASIC
7	n2	n4	0	5	0	X24	.	.	5.00000	NONKEY ARC BASIC
8	t	s	-1	99999999	0	XTS	.	.	0.00000	LOWERBD NONBASIC
9	n3	t	0	4	0	X3T	.	99999998	3.60000	NONKEY ARC BASIC
10	n4	t	0	7	0	X4T	.	99999998	6.40000	KEY_ARC BASIC
11			0	10	0	G12	.	.	0.00000	LOWERBD NONBASIC
12			0	5	0	G24	.	.	5.00000	UPPERBD NONBASIC
13			0	1	0	G3T	.	.	0.00000	LOWERBD NONBASIC
14			0	10	0	G43	.	.	3.20000	NONKEY BASIC
15			0	9	0	GS2	.	.	7.33333	NONKEY BASIC

A.8.3 Network 4

```
title 'Network 4';
title3 'Nodes for Network 4';
   data noded;
      input _node_ $ _sd_;
      cards;
   n1 0
   n6 0
;

title3 'Arcs for Network 4';
   data arcd1;
      input _from_ $ _to_ $ _cost_ _capac_ _lo_ _name_ $;
      cards;
   n1 n2a 0 500 . x12a
   n1 n3a 0 300 . x13a
   n1 n4a 0 600 . x14a
   n1 n5a 0 5 . x15a
   n2a n2b 0 500 . x2a2b
   n2a n3a 0 5 . x2a3a
   n2b n5a 0 500 . x2b5a
   n3a n3b 0 300 . x3a3b
   n3a n4a 0 5 . x3a4a
   n3b n5a 0 200 . x3b5a
   n4a n4b 0 300 . x4a4b
   n4b n6 0 300 . x4b6
   n5a n5b 0 600 . x5a5b
   n5b n6 0 350 . x5b6
   n6 n1 -1 . . x61
   . . 0 10 . g15a
   . . 0 7 . g2a2b
   . . 0 10 . g2a3a
   . . 0 4 . g2b5a
   . . 0 3 . g3a3b
   . . 0 10 . g3a4a
   . . 0 7 . g3b5a
   . . 0 5 . g4a4b
   . . 0 4 . g4b6
   . . 0 2 . g5a5b
   . . 0 3 . g5b6
;

title3 'Side constraints';
data cond1;
   input
x15a x2a2b x2a3a x2b5a x3a3b x3a4a x3b5a x4a4b x4b6 x5a5b x5b6
g15a g2a2b g2a3a g2b5a g3a3b g3a4a g3b5a g4a4b g4b6 g5a5b g5b6 _type_ $
_rhs_;
   cards;
1 . . . . . . . . . . -0.5 . . . . . . . . . . LE 0
. 1 . . . . . . . . . . -50 . . . . . . . . . LE 150
. . 1 . . . . . . . . . . -0.5 . . . . . . . . LE 0
. . . 1 . . . . . . . . . . -50 . . . . . . . LE 300
. . . . 1 . . . . . . . . . . -30 . . . . . . L3 210
. . . . . 1 . . . . . . . . . . -0.5 . . . . . LE 0
. . . . . . 1 . . . . . . . . . . -20 . . . . LE 60
. . . . . . . 1 . . . . . . . . . . -30 . . . LE 150
. . . . . . . . 1 . . . . . . . . . . -30 . . LE 180
. . . . . . . . . 1 . . . . . . . . . . -60 . LE 480
. . . . . . . . . . 1 . . . . . . . . . . -35 LE 245
. . . . . . . . . . . 50 50 50 50 50 50 50 50 50 50 50 LE 1000
;
```

```
proc netflow
  nodedata=noded
  arcdata=arcd1
  condata=cond1
  conout=solution
  maxflow
  sourcenode=n1
  sinknode=n6
  namectrl=1;

print problem;

proc print data=solution;
sum _fcost_;
title3 'upper bound after improving reliability ';

NOTE: Number of nodes= 10 .
NOTE: Number of arcs= 15 .
WARNING: The value of the TYPE list variable L3       in obs 5 in CONDATA is unrecognized or is unimplemented. This obs w
         ignored.
NOTE: Number of nonarc variables= 11 .
NOTE: Number of iterations performed (neglecting any constraints)= 12 .
NOTE: Of these, 8 were degenerate.
NOTE: Maximal flow= 650 .
NOTE: Optimum (neglecting any constraints) found.
NOTE: Minimal total cost= 0 .
NOTE: Number of <= side constraints= 11 .
NOTE: Number of == side constraints= 0 .
NOTE: Number of >= side constraints= 0 .
NOTE: Number of arc and nonarc variable side constraint coefficients= 31 .
NOTE: Number of iterations, optimizing with constraints= 11 .
NOTE: Of these, 2 were degenerate.
NOTE: Maximal flow= 650 .
NOTE: Optimum reached.
NOTE: Minimal total cost= 0 .
NOTE: The data set WORK.SOLUTION has 26 observations and 14 variables.
NOTE: The PROCEDURE NETFLOW printed pages 1-3.
```

upper bound after improving reliability

OBS	_FROM_	_TO_	_COST_	_CAPAC_	_LO_	_NAME_	_SUPPLY_	_DEMAND_	_FLOW_	_STATUS_
1	n6	n1	-1	99999999	0	X61	.	.	0.0	LOWERBD NONBASIC
2	n1	n2a	0	500	0	X12A	99999998	.	300.0	KEY_ARC BASIC
3	n2a	n2b	0	500	0	X2A2B	.	.	300.0	KEY_ARC BASIC
4	n1	n3a	0	300	0	X13A	99999998	.	47.5	KEY_ARC BASIC
5	n2a	n3a	0	5	0	X2A3A	.	.	0.0	LOWERBD NONBASIC
6	n3a	n3b	0	300	0	X3A3B	.	.	47.5	KEY_ARC BASIC
8	n3a	n4a	0	5	0	X3A4A	.	.	0.0	LOWERBD NONBASIC
9	n4a	n4b	0	300	0	X4A4B	.	.	300.0	KEY_ARC BASIC
10	n1	n5a	0	5	0	X15A	99999998	.	2.5	NONKEY ARC BASIC
11	n2b	n5a	0	500	0	X2B5A	.	.	300.0	KEY_ARC BASIC
12	n3b	n5a	0	200	0	X3B5A	.	.	47.5	NONKEY ARC BASIC
13	n5a	n5b	0	600	0	X5A5B	.	.	350.0	KEY_ARC BASIC
14	n4b	n6	0	300	0	X4B6	.	99999998	300.0	UPPERBD NONBASIC
15	n5b	n6	0	350	0	X5B6	.	99999998	350.0	NONKEY ARC BASIC
16			0	10	0	G15A	.	.	5.0	NONKEY BASIC
17			0	7	0	G2A2B	.	.	3.0	NONKEY BASIC
18			0	10	0	G2A3A	.	.	0.0	LOWERBD NONBASIC
19			0	4	0	G2B5A	.	.	0.0	LOWERBD NONBASIC
21			0	10	0	G3A4A	.	.	0.0	LOWERBD NONBASIC
22			0	7	0	G3B5A	.	.	0.0	LOWERBD NONBASIC
23			0	5	0	G4A4B	.	.	5.0	NONKEY BASIC
24			0	4	0	G4B6	.	.	4.0	NONKEY BASIC

25	0	2	0	G5A5B	.	.	0.0	LOWERBD	NONBASIC
26	0	3	0	G5B6	.	.	3.0	UPPERBD	NONBASIC

A.9 Network A

A.9.1 Formula Input File

```
/***************************************************************/
/*                          Network A                          */
/***************************************************************/

% Arc relationship

 arc(s,20).
 arc(s,21).
 arc(s,22).
 arc(s,23).
 arc(1,29).
 arc(1,30).
 arc(1,31).
 arc(2,25).
 arc(2,26).
 arc(3,27).
 arc(3,28).
 arc(4,24).
 arc(5,32).
 arc(5,33).
 arc(6,42).
 arc(7,43).
 arc(8,44).
 arc(9,34).
 arc(10,37).
 arc(11,38).
 arc(12,35).
 arc(13,36).
 arc(14,45).
 arc(14,46).
 arc(14,47).
 arc(15,39).
 arc(15,40).
 arc(15,41).
 arc(16,51).
 arc(17,50).
 arc(18,49).
 arc(19,48).
 arc(20,4).
 arc(21,2).
 arc(22,3).
 arc(23,1).
 arc(24,14).
 arc(25,14).
 arc(26,5).
 arc(27,11).
 arc(28,9).
 arc(29,12).
 arc(30,13).
 arc(31,14).
 arc(32,10).
 arc(33,11).
 arc(34,15).
 arc(35,15).
 arc(36,15).
 arc(37,15).
 arc(38,15).
 arc(39,6).
 arc(40,7).
 arc(41,8).
```

```
arc(42,14).
arc(43,14).
arc(44,14).
arc(45,19).
arc(46,18).
arc(47,17).
arc(48,16).
arc(49,16).
arc(50,16).
arc(51,t).

% Component survival probability

prob(s,1).
prob(1,1).
prob(2,0.3).
prob(3,0.7).
prob(4,0.5).
prob(5,0.8).
prob(6,1).
prob(7,0.3).
prob(8,0.7).
prob(9,0.5).
prob(10,0.8).
prob(11,1).
prob(12,0.3).
prob(13,0.7).
prob(14,0.5).
prob(15,0.8).
prob(16,0.8).
prob(17,0.7).
prob(18,0.3).
prob(19,1).
prob(20,1).
prob(21,1).
prob(22,1).
prob(23,1).
prob(24,1).
prob(25,0.6).
prob(26,0.3).
prob(27,1).
prob(28,1).
prob(29,1).
prob(30,1).
prob(31,1).
prob(32,0.6).
prob(33,0.7).
prob(34,1).
prob(35,0.7).
prob(36,1).
prob(37,0.6).
prob(38,1).
prob(39,0.3).
prob(40,0.6).
prob(41,0.7).
prob(42,0.6).
prob(43,0.6).
prob(44,0.3).
prob(45,0.6).
prob(46,0.6).
prob(47,0.3).
prob(48,0.3).
prob(49,0.6).
prob(50,0.7).
prob(51,1).
```

```
prob(t,1).

% Component capacity

cap(1,*).
cap(2,*).
cap(3,*ΞΔ).
cap(4,*).
cap(5,*).
cap(6,*).
cap(7,*).
cap(8,*).
cap(9,*).
cap(10,*).
cap(11,*).
cap(12,*).
cap(13,*).
cap(14,*).
cap(15,*).
cap(16,*).
cap(17,*).
cap(18,*).
cap(19,*).
cap(20,*).
cap(21,*).
cap(22,*).
cap(23,*).
cap(24,1200).
cap(25,1200).
cap(26,1200).
cap(27,1200).
cap(28,1200).
cap(29,1200).
cap(30,1200).
cap(31,1200).
cap(32,1200).
cap(33,1200).
cap(34,4800).
cap(35,4800).
cap(36,4800).
cap(37,4800).
cap(38,4800).
cap(39,4800).
cap(40,4800).
cap(41,4800).
cap(42,4800).
cap(43,4800).
cap(44,4800).
cap(45,4800).
cap(46,4800).
cap(47,4800).
cap(48,4800).
cap(49,4800).
cap(50,4800).
cap(51,*).

% Cost of increasing capacity by one unit

cost(_,100).

% Predetermined lump sum of capacity increase

invest(_,100).

% Budget available
```

```
budget(100000).

/*--------- end ---------*/
```

A.9.2 Paths

```
******************************************************
*  Following is a list of all paths from 's" to "t"    *
*  of the network described in the input data file.   *
******************************************************


  Path 1: s 20 4 24 14 45 19 48 16 51 t
  Reliability: 0.036

  Path 2: s 20 4 24 14 46 18 49 16 51 t
  Reliability: 0.0216

  Path 3: s 20 4 24 14 47 17 50 16 51 t
  Reliability: 0.0294

  Path 4: s 21 2 25 14 45 19 48 16 51 t
  Reliability: 0.01296

  Path 5: s 21 2 25 14 46 18 49 16 51 t
  Reliability: 0.007776

  Path 6: s 21 2 25 14 47 17 50 16 51 t
  Reliability: 0.010584

  Path 7: s 21 2 26 5 32 10 37 15 39 6 42 14 45 19 48 16 51 t
  Reliability: 0.00021499

  Path 8: s 21 2 26 5 32 10 37 15 39 6 42 14 46 18 49 16 51 t
  Reliability: 0.00012899

  Path 9: s 21 2 26 5 32 10 37 15 39 6 42 14 47 17 50 16 51 t
  Reliability: 0.00017558

  Path 10: s 21 2 26 5 32 10 37 15 40 7 43 14 45 19 48 16 51 t
  Reliability: 0.00012899

  Path 11: s 21 2 26 5 32 10 37 15 40 7 43 14 46 18 49 16 51 t
  Reliability: 0.0000774

  Path 12: s 21 2 26 5 32 10 37 15 40 7 43 14 47 17 50 16 51 t
  Reliability: 0.00010535

  Path 13: s 21 2 26 5 32 10 37 15 41 8 44 14 45 19 48 16 51 t
  Reliability: 0.00017558

  Path 14: s 21 2 26 5 32 10 37 15 41 8 44 14 46 18 49 16 51 t
  Reliability: 0.00010535
```

Path 15: s 21 2 26 5 32 10 37 15 41 8 44 14 47 17 50 16 51 t
Reliability: 0.00014339

Path 16: s 21 2 26 5 33 11 38 15 39 6 42 14 45 19 48 16 51 t
Reliability: 0.00052255

Path 17: s 21 2 26 5 33 11 38 15 39 6 42 14 46 18 49 16 51 t
Reliability: 0.00031353

Path 18: s 21 2 26 5 33 11 38 15 39 6 42 14 47 17 50 16 51 t
Reliability: 0.00042675

Path 19: s 21 2 26 5 33 11 38 15 40 7 43 14 45 19 48 16 51 t
Reliability: 0.000313

Path 20: s 21 2 26 5 33 11 38 15 40 7 43 14 46 18 49 16 51 t
Reliability: 0.00018812

Path 21: s 21 2 26 5 33 11 38 15 40 7 43 14 47 17 50 16 51 t
Reliability: 0.00025605

Path 22: s 21 2 26 5 33 11 38 15 41 8 44 14 45 19 48 16 51 t
Reliability: 0.00042675

Path 23: s 21 2 26 5 33 11 38 15 41 8 44 14 46 18 49 16 51 t
Reliability: 0.00025605

Path 24: s 21 2 26 5 33 11 38 15 41 8 44 14 47 17 50 16 51 t
Reliability: 0.00034851

Path 25: s 22 3 27 11 38 15 39 6 42 14 45 19 48 16 51 t
Reliability: 0.0072576

Path 26: s 22 3 27 11 38 15 39 6 42 14 46 18 49 16 51 t
Reliability: 0.00435456

Path 27: s 22 3 27 11 38 15 39 6 42 14 47 17 50 16 51 t
Reliability: 0.00592704

Path 28: s 22 3 27 11 38 15 40 7 43 14 45 19 48 16 51 t
Reliability: 0.00435456

Path 29: s 22 3 27 11 38 15 40 7 43 14 46 18 49 16 51 t
Reliability: 0.00261274

Path 30: s 22 3 27 11 38 15 40 7 43 14 47 17 50 16 51 t
Reliability: 0.00355622

Path 31: s 22 3 27 11 38 15 41 8 44 14 45 19 48 16 51 t
Reliability: 0.00592704

Path 32: s 22 3 27 11 38 15 41 8 44 14 46 18 49 16 51 t
Reliability: 0.00355622

Path 33: s 22 3 27 11 38 15 41 8 44 14 47 17 50 16 51 t
Reliability: 0.00484042

Path 34: s 22 3 28 9 34 15 39 6 42 14 45 19 48 16 51 t
Reliability: 0.0036288

Path 35: s 22 3 28 9 34 15 39 6 42 14 46 18 49 16 51 t
Reliability: 0.00217728

Path 36: s 22 3 28 9 34 15 39 6 42 14 47 17 50 16 51 t
Reliability: 0.00296352

Path 37: s 22 3 28 9 34 15 40 7 43 14 45 19 48 16 51 t
Reliability: 0.00217728

Path 38: s 22 3 28 9 34 15 40 7 43 14 46 18 49 16 51 t
Reliability: 0.00130637

Path 39: s 22 3 28 9 34 15 40 7 43 14 47 17 50 16 51 t
Reliability: 0.00177811

Path 40: s 22 3 28 9 34 15 41 8 44 14 45 19 48 16 51 t
Reliability: 0.00296352

Path 41: s 22 3 28 9 34 15 41 8 44 14 46 18 49 16 51 t
Reliability: 0.00177811

Path 42: s 22 3 28 9 34 15 41 8 44 14 47 17 50 16 51 t
Reliability: 0.00242021

Path 43: s 23 1 29 12 35 15 39 6 42 14 45 19 48 16 51 t
Reliability: 0.00217728

Path 44: s 23 1 29 12 35 15 39 6 42 14 46 18 49 16 51 t
Reliability: 0.00130637

Path 45: s 23 1 29 12 35 15 39 6 42 14 47 17 50 16 51 t
Reliability: 0.00177811

Path 46: s 23 1 29 12 35 15 40 7 43 14 45 19 48 16 51 t
Reliability: 0.00130637

Path 47: s 23 1 29 12 35 15 40 7 43 14 46 18 49 16 51 t
Reliability: 0.00078382

Path 48: s 23 1 29 12 35 15 40 7 43 14 47 17 50 16 51 t
Reliability: 0.00106687

```
Path 49: s 23 1 29 12 35 15 41 8 44 14 45 19 48 16 51 t
Reliability: 0.00177811

Path 50: s 23 1 29 12 35 15 41 8 44 14 46 18 49 16 51 t
Reliability: 0.00106687

Path 51: s 23 1 29 12 35 15 41 8 44 14 47 17 50 16 51 t
Reliability: 0.00145212

Path 52: s 23 1 30 13 36 15 39 6 42 14 45 19 48 16 51 t
Reliability: 0.0072576

Path 53: s 23 1 30 13 36 15 39 6 42 14 46 18 49 16 51 t
Reliability: 0.00435456

Path 54: s 23 1 30 13 36 15 39 6 42 14 47 17 50 16 51 t
Reliability: 0.00592704

Path 55: s 23 1 30 13 36 15 40 7 43 14 45 19 48 16 51 t
Reliability: 0.00435456

Path 56: s 23 1 30 13 36 15 40 7 43 14 46 18 49 16 51 t
Reliability: 0.00261274

Path 57: s 23 1 30 13 36 15 40 7 43 14 47 17 50 16 51 t
Reliability: 0.00355622

Path 58: s 23 1 30 13 36 15 41 8 44 14 45 19 48 16 51 t
Reliability: 0.00592704

Path 59: s 23 1 30 13 36 15 41 8 44 14 46 18 49 16 51 t
Reliability: 0.00355622

Path 60: s 23 1 30 13 36 15 41 8 44 14 47 17 50 16 51 t
Reliability: 0.00484042

Path 61: s 23 1 31 14 45 19 48 16 51 t
Reliability: 0.072

Path 62: s 23 1 31 14 46 18 49 16 51 t
Reliability: 0.0432

Path 63: s 23 1 31 14 47 17 50 16 51 t
Reliability: 0.0588

* ------ end ------ *
```

A.9.3 Lower Bound For Original Network

```
LP83 a.lb output a.lbr

  Copyright (C) 1986 by Sunset Software.
  All Rights Reserved Worldwide.
  1613 Chelsea Road,  Suite 153
  San Marino, California  91108   U.S.A.

  Licensed Solely To: Wright-Patterson Air Force Base

 ..Title

 Lower Bound Formulation

 ..Objective Maximize

 0.036 f1 + 0.0216 f2 + 0.0294 f3 + 0.01296 f4 +
 0.007776 f5 + 0.010584 f6 + 0.00021499 f7 + 0.00012899 f8 +
 0.00017558 f9 + 0.00012899 f10 + 0.0000774 f11 + 0.00010535 f12 +
 0.00017558 f13 + 0.00010535 f14 + 0.00014339 f15 + 0.00052255 f16 +
 0.00031353 f17 + 0.00042675 f18 + 0.00031353 f19 + 0.00018812 f20 +
 0.00025605 f21 + 0.00042675 f22 + 0.00025605 f23 + 0.00034851 f24 +
 0.0072576 f25 + 0.00435456 f26 + 0.00592704 f27 + 0.00435456 f28 +
 0.00261274 f29 + 0.00355622 f30 + 0.00592704 f31 + 0.00355622 f32 +
 0.00484042 f33 + 0.0036288 f34 + 0.00217728 f35 + 0.00296352 f36 +
 0.00217728 f37 + 0.00130637 f38 + 0.00177811 f39 + 0.00296352 f40 +
 0.00177811 f41 + 0.00242021 f42 + 0.00217728 f43 + 0.00130637 f44 +
 0.00177811 f45 + 0.00130637 f46 + 0.00078382 f47 + 0.00106687 f48 +
 0.00177811 f49 + 0.00106687 f50 + 0.00145212 f51 + 0.0072576 f52 +
 0.00435456 f53 + 0.00592704 f54 + 0.00435456 f55 + 0.00261274 f56 +
 0.00355622 f57 + 0.00592704 f58 + 0.00355622 f59 + 0.00484042 f60 +
 0.072 f61 + 0.0432 f62 + 0.0588 f63

 ..Constraints

 Arc 24: f1 + f2 + f3 <=  1200

 Arc 25: f4 + f5 + f6 <=  1200

 Arc 26: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
         f14 + f15 + f16 + f17 + f18 + f19 + f20 +
         f21 + f22 + f23 + f24 <=  1200

 Arc 27: f25 + f26 + f27 + f28 + f29 + f30 + f31 +
         f32 + f33 <=  1200

 Arc 28: f34 + f35 + f36 + f37 + f38 + f39 + f40 +
         f41 + f42 <=  1200

 Arc 29: f43 + f44 + f45 + f46 + f47 + f48 + f49 +
```

```
      f50 + f51 <=  1200

Arc 30: f52 + f53 + f54 + f55 + f56 + f57 + f58 +
        f59 + f60 <=  1200

Arc 31: f61 + f62 + f63 <=  1200

Arc 32: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
        f14 + f15 <=  1200

Arc 33: f16 + f17 + f18 + f19 + f20 + f21 + f22 +
        f23 + f24 <=  1200

Arc 34: f34 + f35 + f36 + f37 + f38 + f39 + f40 +
        f41 + f42 <=  4800

Arc 35: f43 + f44 + f45 + f46 + f47 + f48 + f49 +
        f50 + f51 <=  4800

Arc 36: f52 + f53 + f54 + f55 + f56 + f57 + f58 +
        f59 + f60 <=  4800

Arc 37: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
        f14 + f15 <=  4800

Arc 38: f16 + f17 + f18 + f19 + f20 + f21 + f22 +
        f23 + f24 + f25 + f26 + f27 + f28 + f29 +
        f30 + f31 + f32 + f33 <=  4800

Arc 39: f7 + f8 + f9 + f16 + f17 + f18 + f25 +
        f26 + f27 + f34 + f35 + f36 + f43 + f44 +
        f45 + f52 + f53 + f54 <=  4800

Arc 40: f10 + f11 + f12 + f19 + f20 + f21 + f28 +
        f29 + f30 + f37 + f38 + f39 + f46 + f47 +
        f48 + f55 + f56 + f57 <=  4800

Arc 41: f13 + f14 + f15 + f22 + f23 + f24 + f31 +
        f32 + f33 + f40 + f41 + f42 + f49 + f50 +
        f51 + f58 + f59 + f60 <=  4800

Arc 42: f7 + f8 + f9 + f16 + f17 + f18 + f25 +
        f26 + f27 + f34 + f35 + f36 + f43 + f44 +
        f45 + f52 + f53 + f54 <=  4800

Arc 43: f10 + f11 + f12 + f19 + f20 + f21 + f28 +
        f29 + f30 + f37 + f38 + f39 + f46 + f47 +
        f48 + f55 + f56 + f57 <=  4800

Arc 44: f13 + f14 + f15 + f22 + f23 + f24 + f31 +
        f32 + f33 + f40 + f41 + f42 + f49 + f50 +
```

```
        f51 + f58 + f59 + f60 <=  4800

Arc 45: f1 + f4 + f7 + f10 + f13 + f16 + f19 +
        f22 + f25 + f28 + f31 + f34 + f37 + f40 +
        f43 + f46 + f49 + f52 + f55 + f58 + f61 <=  4800

Arc 46: f2 + f5 + f8 + f11 + f14 + f17 + f20 +
        f23 + f26 + f29 + f32 + f35 + f38 + f41 +
        f44 + f47 + f50 + f53 + f56 + f59 + f62 <=  4800

Arc 47: f3 + f6 + f9 + f12 + f15 + f18 + f21 +
        f24 + f27 + f30 + f33 + f36 + f39 + f42 +
        f45 + f48 + f51 + f54 + f57 + f60 + f63 <=  4800

Arc 48: f1 + f4 + f7 + f10 + f13 + f16 + f19 +
        f22 + f25 + f28 + f31 + f34 + f37 + f40 +
        f43 + f46 + f49 + f52 + f55 + f58 + f61 <=  4800

Arc 49: f2 + f5 + f8 + f11 + f14 + f17 + f20 +
        f23 + f26 + f29 + f32 + f35 + f38 + f41 +
        f44 + f47 + f50 + f53 + f56 + f59 + f62 <=  4800

Arc 50: f3 + f6 + f9 + f12 + f15 + f18 + f21 +
        f24 + f27 + f30 + f33 + f36 + f39 + f42 +
        f45 + f48 + f51 + f54 + f57 + f60 + f63 <=  4800

* ------ end ------ *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  392K bytes.
  Objective Function is MAXIMIZED.
  Variables:        63
  Constraints:      27
    27 LE,   0 EQ,   0 GE.
  Non-zero LP elements:     369
  Disk Space:     0K bytes.
  Page Space:    14K bytes.
  Capacity:  9.8% used.
  Estimated Time: 00:01:01

Iter   12
Solution Time:  00:00:03
A L T E R N A T E   S O L U T I O N S
```

File: A   8/28/90 07:39:26 Page 1-1

SOLUTION (Maximized): 167.0817 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	1,200.0000	0.0360	f2	0.0000	0.0216
f3	0.0000	0.0294	f4	1,200.0000	0.0130
f5	0.0000	0.0078	f6	0.0000	0.0106
f7	0.0000	0.0002	f8	0.0000	0.0001
f9	0.0000	0.0002	f10	0.0000	0.0001
f11	0.0000	0.0001	f12	0.0000	0.0001
f13	0.0000	0.0002	f14	0.0000	0.0001
f15	0.0000	0.0001	f16	0.0000	0.0005
f17	0.0000	0.0003	f18	0.0000	0.0004
f19	0.0000	0.0003	f20	0.0000	0.0002

File: A 8/28/90 07:39:26 Page 1-2

SOLUTION (Maximized): 167.0817 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f21	0.0000	0.0003	f22	0.0000	0.0004
f23	0.0000	0.0003	f24	1,200.0000	0.0003
f25	1,200.0000	0.0073	f26	0.0000	0.0044
f27	0.0000	0.0059	f28	0.0000	0.0044
f29	0.0000	0.0026	f30	0.0000	0.0036
f31	0.0000	0.0059	f32	0.0000	0.0036
f33	0.0000	0.0048	f34	0.0000	0.0036
f35	0.0000	0.0022	f36	1,200.0000	0.0030
f37	0.0000	0.0022	f38	0.0000	0.0013
f39	0.0000	0.0018	f40	0.0000	0.0030

File: A 8/28/90 07:39:26 Page 1-3

SOLUTION (Maximized): 167.0817 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f41	0.0000	0.0018	f42	0.0000	0.0024
f43	0.0000	0.0022	f44	0.0000	0.0013
f45	1,200.0000	0.0018	f46	0.0000	0.0013
f47	0.0000	0.0008	f48	0.0000	0.0011
f49	0.0000	0.0018	f50	0.0000	0.0011
f51	0.0000	0.0015	f52	0.0000	0.0073
f53	0.0000	0.0044	f54	1,200.0000	0.0059
f55	0.0000	0.0044	f56	0.0000	0.0026
f57	0.0000	0.0036	f58	0.0000	0.0059
f59	0.0000	0.0036	f60	0.0000	0.0048

File: A 8/28/90 07:39:26 Page 1-4

SOLUTION (Maximized): 167.0817 Lower Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f61	1,200.0000	0.0720	f62	0.0000	0.0432
f63	0.0000	0.0588			

File: A 8/28/90 07:39:26 Page 1-5

CONSTRAINTS: Lower Bound Formulation

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 24	1,200.0000	<	1,200.0000	Arc 25	1,200.0000	<	1,200.0000
Arc 26	1,200.0000	<	1,200.0000	Arc 27	1,200.0000	<	1,200.0000
Arc 28	1,200.0000	<	1,200.0000	Arc 29	1,200.0000	<	1,200.0000
Arc 30	1,200.0000	<	1,200.0000	Arc 31	1,200.0000	<	1,200.0000
Arc 32	0.0000	<	1,200.0000	Arc 33	1,200.0000	<	1,200.0000
Arc 34	1,200.0000	<	4,800.0000	Arc 35	1,200.0000	<	4,800.0000
Arc 36	1,200.0000	<	4,800.0000	Arc 37	0.0000	<	4,800.0000
Arc 38	2,400.0000	<	4,800.0000	Arc 39	4,800.0000	<	4,800.0000
Arc 40	0.0000	<	4,800.0000	Arc 41	1,200.0000	<	4,800.0000
Arc 42	4,800.0000	<	4,800.0000	Arc 43	0.0000	<	4,800.0000

File: A 8/28/90 07:39:26 Page 1-6

CONSTRAINTS: Lower Bound Formulation

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 44	1,200.0000	<	4,800.0000	Arc 45	4,800.0000	<	4,800.0000
Arc 46	0.0000	<	4,800.0000	Arc 47	4,800.0000	<	4,800.0000
Arc 48	4,800.0000	<	4,800.0000	Arc 49	0.0000	<	4,800.0000
Arc 50	4,800.0000	<	4,800.0000				

Total Error: 0.000000

A.9.4 Upper Bound For Original Network

```
LP83 a.ub output a.ubr

 Copyright (C) 1986 by Sunset Software.
 All Rights Reserved Worldwide.
 1613 Chelsea Road,  Suite 153
 San Marino, California  91108   U.S.A.

 Licensed Solely To: Wright-Patterson Air Force Base

..Title

Upper Bound Formulation

..Objective Maximize

 f1 +  f2 +  f3 +  f4 +
 f5 +  f6 +  f7 +  f8 +
 f9 +  f10 +  f11 +  f12 +
 f13 +  f14 +  f15 +  f16 +
 f17 +  f18 +  f19 +  f20 +
 f21 +  f22 +  f23 +  f24 +
 f25 +  f26 +  f27 +  f28 +
 f29 +  f30 +  f31 +  f32 +
 f33 +  f34 +  f35 +  f36 +
 f37 +  f38 +  f39 +  f40 +
 f41 +  f42 +  f43 +  f44 +
 f45 +  f46 +  f47 +  f48 +
 f49 +  f50 +  f51 +  f52 +
 f53 +  f54 +  f55 +  f56 +
 f57 +  f58 +  f59 +  f60 +
 f61 +  f62 +  f63

..Constraints

Arc 24: f1 + f2 + f3 <=  1200

Arc 25: f4 + f5 + f6 <=  720.0

Arc 26: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
        f14 + f15 + f16 + f17 + f18 + f19 + f20 +
        f21 + f22 + f23 + f24 <=  360.0

Arc 27: f25 + f26 + f27 + f28 + f29 + f30 + f31 +
        f32 + f33 <=  1200

Arc 28: f34 + f35 + f36 + f37 + f38 + f39 + f40 +
        f41 + f42 <=  1200

Arc 29: f43 + f44 + f45 + f46 + f47 + f48 + f49 +
```

```
        f50 + f51 <=  1200

Arc 30: f52 + f53 + f54 + f55 + f56 + f57 + f58 +
        f59 + f60 <=  1200

Arc 31: f61 + f62 + f63 <=  1200

Arc 32: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
        f14 + f15 <=  720.0

Arc 33: f16 + f17 + f18 + f19 + f20 + f21 + f22 +
        f23 + f24 <=  840.0

Arc 34: f34 + f35 + f36 + f37 + f38 + f39 + f40 +
        f41 + f42 <=  4800

Arc 35: f43 + f44 + f45 + f46 + f47 + f48 + f49 +
        f50 + f51 <=  3360.0

Arc 36: f52 + f53 + f54 + f55 + f56 + f57 + f58 +
        f59 + f60 <=  4800

Arc 37: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
        f14 + f15 <=  2880.0

Arc 38: f16 + f17 + f18 + f19 + f20 + f21 + f22 +
        f23 + f24 + f25 + f26 + f27 + f28 + f29 +
        f30 + f31 + f32 + f33 <=  4800

Arc 39: f7 + f8 + f9 + f16 + f17 + f18 + f25 +
        f26 + [illegible] f34 + f35 + f36 + f43 + f44 +
        f45 + f52 + f53 + f54 <=  1440.0

Arc 40: f10 + f11 + f12 + f19 + f20 + f21 + f28 +
        f29 + f30 + f37 + f38 + f39 + f46 + f47 +
        f48 + f55 + f56 + f57 <=  2880.0

Arc 41: f13 + f14 + f15 + f22 + f23 + f24 + f31 +
        f32 + f33 + f40 + f41 + f42 + f49 + f50 +
        f51 + f58 + f59 + f60 <=  3360.0

Arc 42: f7 + f8 + f9 + f16 + f17 + f18 + f25 +
        f26 + f27 + f34 + f35 + f36 + f43 + f44 +
        f45 + f52 + f53 + f54 <=  2880.0

Arc 43: f10 + f11 + f12 + f19 + f20 + f21 + f28 +
        f29 + f30 + f37 + f38 + f39 + f46 + f47 +
        f48 + f55 + f56 + f57 <=  2880.0

Arc 44: f13 + f14 + f15 + f22 + f23 + f24 + f31 +
        f32 + f33 + f40 + f41 + f42 + f49 + f50 +
```

```
          f51 + f58 + f59 + f60 <=  1440.0

 Arc 45: f1 + f4 + f7 + f10 + f13 + f16 + f19 +
         f22 + f25 + f28 + f31 + f34 + f37 + f40 +
         f43 + f46 + f49 + f52 + f55 + f58 + f61 <=  2880.0

 Arc 46: f2 + f5 + f8 + f11 + f14 + f17 + f20 +
         f23 + f26 + f29 + f32 + f35 + f38 + f41 +
         f44 + f47 + f50 + f53 + f56 + f59 + f62 <=  2880.0

 Arc 47: f3 + f6 + f9 + f12 + f15 + f18 + f21 +
         f24 + f27 + f30 + f33 + f36 + f39 + f42 +
         f45 + f48 + f51 + f54 + f57 + f60 + f63 <=  1440.0

 Arc 48: f1 + f4 + f7 + f10 + f13 + f16 + f19 +
         f22 + f25 + f28 + f31 + f34 + f37 + f40 +
         f43 + f46 + f49 + f52 + f55 + f58 + f61 <=  1440.0

 Arc 49: f2 + f5 + f8 + f11 + f14 + f17 + f20 +
         f23 + f26 + f29 + f32 + f35 + f38 + f41 +
         f44 + f47 + f50 + f53 + f56 + f59 + f62 <=  2880.0

 Arc 50: f3 + f6 + f9 + f12 + f15 + f18 + f21 +
         f24 + f27 + f30 + f33 + f36 + f39 + f42 +
         f45 + f48 + f51 + f54 + f57 + f60 + f63 <=  3360.0

* ------ end ------ *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  392K bytes.
  Objective Function is MAXIMIZED.
  Variables:        63
  Constraints:      27
    27 LE,    0 EQ,    0 GE.
  Non-zero LP elements:     369
  Disk Space:     0K bytes.
  Page Space:    14K bytes.
  Capacity:  9.8% used.
  Estimated Time: 00:01:01

Iter    9
Solution Time:  00:00:03
A L T E R N A T E   S O L U T I O N S
```

SOLUTION (Maximized): 5,760.0000 Upper Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f1	0.0000	1.0000	f2	1,200.0000	1.0000
f3	0.0000	1.0000	f4	0.0000	1.0000
f5	720.0000	1.0000	f6	0.0000	1.0000
f7	120.0000	1.0000	f8	120.0000	1.0000
f9	0.0000	1.0000	f10	120.0000	1.0000
f11	0.0000	1.0000	f12	0.0000	1.0000
f13	0.0000	1.0000	f14	0.0000	1.0000
f15	0.0000	1.0000	f16	0.0000	1.0000
f17	0.0000	1.0000	f18	0.0000	1.0000
f19	0.0000	1.0000	f20	0.0000	1.0000

File: A 8/28/90 07:40:25 Page 1-2

SOLUTION (Maximized): 5,760.0000 Upper Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f21	0.0000	1.0000	f22	0.0000	1.0000
f23	0.0000	1.0000	f24	0.0000	1.0000
f25	0.0000	1.0000	f26	840.0000	1.0000
f27	360.0000	1.0000	f28	0.0000	1.0000
f29	0.0000	1.0000	f30	0.0000	1.0000
f31	0.0000	1.0000	f32	0.0000	1.0000
f33	0.0000	1.0000	f34	0.0000	1.0000
f35	0.0000	1.0000	f36	0.0000	1.0000
f37	0.0000	1.0000	f38	0.0000	1.0000
f39	1,080.0000	1.0000	f40	0.0000	1.0000

File: A 8/28/90 07:40:25 Page 1-3

SOLUTION (Maximized): 5,760.0000 Upper Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f41	0.0000	1.0000	f42	0.0000	1.0000
f43	0.0000	1.0000	f44	0.0000	1.0000
f45	0.0000	1.0000	f46	0.0000	1.0000
f47	0.0000	1.0000	f48	0.0000	1.0000
f49	0.0000	1.0000	f50	0.0000	1.0000
f51	0.0000	1.0000	f52	0.0000	1.0000
f53	0.0000	1.0000	f54	0.0000	1.0000
f55	0.0000	1.0000	f56	0.0000	1.0000
f57	0.0000	1.0000	f58	0.0000	1.0000
f59	0.0000	1.0000	f60	0.0000	1.0000

File: A 8/28/90 07:40:25 Page 1-4

SOLUTION (Maximized): 5,760.0000 Upper Bound Formulation

Variable	Activity	Cost	Variable	Activity	Cost
f61	1,200.0000	1.0000	f62	0.0000	1.0000
f63	0.0000	1.0000			

File: A 8/28/90 07:40:25 Page 1-5

CONSTRAINTS: Upper Bound Formulation

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 24	1,200.0000	<	1,200.0000	Arc 25	720.0000	<	720.0000
Arc 26	360.0000	<	360.0000	Arc 27	1,200.0000	<	1,200.0000
Arc 28	1,080.0000	<	1,200.0000	Arc 29	0.0000	<	1,200.0000
Arc 30	0.0000	<	1,200.0000	Arc 31	1,200.0000	<	1,200.0000
Arc 32	360.0000	<	720.0000	Arc 33	0.0000	<	840.0000
Arc 34	1,080.0000	<	4,800.0000	Arc 35	0.0000	<	3,360.0000
Arc 36	0.0000	<	4,800.0000	Arc 37	360.0000	<	2,880.0000
Arc 38	1,200.0000	<	4,800.0000	Arc 39	1,440.0000	<	1,440.0000
Arc 40	1,200.0000	<	2,880.0000	Arc 41	0.0000	<	3,360.0000
Arc 42	1,440.0000	<	2,880.0000	Arc 43	1,200.0000	<	2,880.0000

File: A 8/28/90 07:40:25 Page 1-6

CONSTRAINTS: Upper Bound Formulation

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 44	0.0000	<	1,440.0000	Arc 45	1,440.0000	<	2,880.0000
Arc 46	2,880.0000	<	2,880.0000	Arc 47	1,440.0000	<	1,440.0000
Arc 48	1,440.0000	<	1,440.0000	Arc 49	2,880.0000	<	2,880.0000
Arc 50	1,440.0000	<	3,360.0000				

Total Error: 0.000000

A.9.5 Investment Strategy Model 1

```
LP83 output5.lp output a.is1

 Copyright (C) 1986 by Sunset Software.
 All Rights Reserved Worldwide.
 1613 Chelsea Road,  Suite 153
 San Marino, California  91108   U.S.A.

 Licensed Solely To: Wright-Patterson Air Force Base

..Title

Investment Strategy Model 1

..Objective Maximize

0.036 f1 + 0.0216 f2 + 0.0294 f3 + 0.01296 f4 +
0.007776 f5 + 0.010584 f6 + 0.00021499 f7 + 0.00012899 f8 +
0.00017558 f9 + 0.00012899 f10 + 0.0000774 f11 + 0.00010535 f12 +
0.00017558 f13 + 0.00010535 f14 + 0.00014339 f15 + 0.00052255 f16 +
0.00031353 f17 + 0.00042675 f18 + 0.00031353 f19 + 0.00018812 f20 +
0.00025605 f21 + 0.00042675 f22 + 0.00025605 f23 + 0.00034851 f24 +
0.0072576 f25 + 0.00435456 f26 + 0.00592704 f27 + 0.00435456 f28 +
0.00261274 f29 + 0.00355622 f30 + 0.00592704 f31 + 0.00355622 f32 +
0.00484042 f33 + 0.0036288 f34 + 0.00217728 f35 + 0.00296352 f36 +
0.00217728 f37 + 0.00130637 f38 + 0.00177811 f39 + 0.00296352 f40 +
0.00177811 f41 + 0.00242021 f42 + 0.00217728 f43 + 0.00130637 f44 +
0.00177811 f45 + 0.00130637 f46 + 0.00078382 f47 + 0.00106687 f48 +
0.00177811 f49 + 0.00106687 f50 + 0.00145212 f51 + 0.0072576 f52 +
0.00435456 f53 + 0.00592704 f54 + 0.00435456 f55 + 0.00261274 f56 +
0.00355622 f57 + 0.00592704 f58 + 0.00355622 f59 + 0.00484042 f60 +
0.072 f61 + 0.0432 f62 + 0.0588 f63 +
0 d24 + 0 d25 + 0 d26 + 0 d27 + 0 d28 + 0 d29 + 0 d30 +
0 d31 + 0 d32 + 0 d33 + 0 d34 + 0 d35 + 0 d36 + 0 d37 +
0 d38 + 0 d39 + 0 d40 + 0 d41 + 0 d42 + 0 d43 + 0 d44 +
0 d45 + 0 d46 + 0 d47 + 0 d48 + 0 d49 + 0 d50

..Constraints

Arc 24: f1 + f2 + f3 - d24 <=  1200

Arc 25: f4 + f5 + f6 - d25 <=  1200

Arc 26: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
        f14 + f15 + f16 + f17 + f18 + f19 + f20 +
        f21 + f22 + f23 + f24 - d26 <=  1200

Arc 27: f25 + f26 + f27 + f28 + f29 + f30 + f31 +
        f32 + f33 - d27 <=  1200
```

```
Arc 28: f34 + f35 + f36 + f37 + f38 + f39 + f40 +
        f41 + f42 - d28 <=  1200

Arc 29: f43 + f44 + f45 + f46 + f47 + f48 + f49 +
        f50 + f51 - d29 <=  1200

Arc 30: f52 + f53 + f54 + f55 + f56 + f57 + f58 +
        f59 + f60 - d30 <=  1200

Arc 31: f61 + f62 + f63 - d31 <=  1200

Arc 32: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
        f14 + f15 - d32 <=  1200

Arc 33: f16 + f17 + f18 + f19 + f20 + f21 + f22 +
        f23 + f24 - d33 <=  1200

Arc 34: f34 + f35 + f36 + f37 + f38 + f39 + f40 +
        f41 + f42 - d34 <=  4800

Arc 35: f43 + f44 + f45 + f46 + f47 + f48 + f49 +
        f50 + f51 - d35 <=  4800

Arc 36: f52 + f53 + f54 + f55 + f56 + f57 + f58 +
        f59 + f60 - d36 <=  4800

Arc 37: f7 + f8 + f9 + f10 + f11 + f12 + f13 +
        f14 + f15 - d37 <=  4800

Arc 38: f16 + f17 + f18 + f19 + f20 + f21 + f22 +
        f23 + f24 + f25 + f26 + f27 + f28 + f29 +
        f30 + f31 + f32 + f33 - d38 <=  4800

Arc 39: f7 + f8 + f9 + f16 + f17 + f18 + f25 +
        f26 + f27 + f34 + f35 + f36 + f43 + f44 +
        f45 + f52 + f53 + f54 - d39 <=  4800

Arc 40: f10 + f11 + f12 + f19 + f20 + f21 + f28 +
        f29 + f30 + f37 + f38 + f39 + f46 + f47 +
        f48 + f55 + f56 + f57 - d40 <=  4800

Arc 41: f13 + f14 + f15 + f22 + f23 + f24 + f31 +
        f32 + f33 + f40 + f41 + f42 + f49 + f50 +
        f51 + f58 + f59 + f60 - d41 <=  4800

Arc 42: f7 + f8 + f9 + f16 + f17 + f18 + f25 +
        f26 + f27 + f34 + f35 + f36 + f43 + f44 +
        f45 + f52 + f53 + f54 - d42 <=  4800

Arc 43: f10 + f11 + f12 + f19 + f20 + f21 + f28 +
        f29 + f30 + f37 + f38 + f39 + f46 + f47 +
```

```
        f48 + f55 + f56 + f57 - d43 <=  4800

Arc 44: f13 + f14 + f15 + f22 + f23 + f24 + f31 +
        f32 + f33 + f40 + f41 + f42 + f49 + f50 +
        f51 + f58 + f59 + f60 - d44 <=  4800

Arc 45: f1 + f4 + f7 + f10 + f13 + f16 + f19 +
        f22 + f25 + f28 + f31 + f34 + f37 + f40 +
        f43 + f46 + f49 + f52 + f55 + f58 + f61 - d45 <=  4800

Arc 46: f2 + f5 + f8 + f11 + f14 + f17 + f20 +
        f23 + f26 + f29 + f32 + f35 + f38 + f41 +
        f44 + f47 + f50 + f53 + f56 + f59 + f62 - d46 <=  4800

Arc 47: f3 + f6 + f9 + f12 + f15 + f18 + f21 +
        f24 + f27 + f30 + f33 + f36 + f39 + f42 +
        f45 + f48 + f51 + f54 + f57 + f60 + f63 - d47 <=  4800

Arc 48: f1 + f4 + f7 + f10 + f13 + f16 + f19 +
        f22 + f25 + f28 + f31 + f34 + f37 + f40 +
        f43 + f46 + f49 + f52 + f55 + f58 + f61 - d48 <=  4800

Arc 49: f2 + f5 + f8 + f11 + f14 + f17 + f20 +
        f23 + f26 + f29 + f32 + f35 + f38 + f41 +
        f44 + f47 + f50 + f53 + f56 + f59 + f62 - d49 <=  4800

Arc 50: f3 + f6 + f9 + f12 + f15 + f18 + f21 +
        f24 + f27 + f30 + f33 + f36 + f39 + f42 +
        f45 + f48 + f51 + f54 + f57 + f60 + f63 - d50 <=  4800

Budget: 100 d24 + 100 d25 + 100 d26 + 100 d27 + 100 d28 + 100 d29 +
        100 d30 + 100 d31 + 100 d32 + 100 d33 + 100 d34 +
        100 d35 + 100 d36 + 100 d37 + 100 d38 + 100 d39 +
        100 d40 + 100 d41 + 100 d42 + 100 d43 + 100 d44 +
        100 d45 + 100 d46 + 100 d47 + 100 d48 + 100 d49 +
        100 d50  <= 100000

* ------ end ------ *



Statistics-
  LP83 Version 5.00a
  Machine memory:  640K bytes.
  Pagable memory:  392K bytes.
  Objective Function is MAXIMIZED.
  Variables:        90
  Constraints:      28
    28 LE,   0 EQ,   0 GE.
  Non-zero LP elements:     423
```

Disk Space: 0K bytes.
Page Space: 20K bytes.
Capacity: 12.1% used.
Estimated Time: 00:01:40

Iter 14
Solution Time: 00:00:04
A L T E R N A T E S O L U T I O N S

File: Output5 10/03/90 09:44:12 Page 1-1
SOLUTION (Maximized): 237.6587 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
f1	1,200.0000	0.0360	f2	0.0000	0.0216
f3	0.0000	0.0294	f4	1,200.0000	0.0130
f5	0.0000	0.0078	f6	0.0000	0.0106
f7	0.0000	0.0002	f8	0.0000	0.0001
f9	0.0000	0.0002	f10	0.0000	0.0001
f11	0.0000	0.0001	f12	0.0000	0.0001
f13	0.0000	0.0002	f14	0.0000	0.0001
f15	0.0000	0.0001	f16	0.0000	0.0005
f17	0.0000	0.0003	f18	0.0000	0.0004
f19	0.0000	0.0003	f20	0.0000	0.0002

File: Output5 10/03/90 09:44:12 Page 1-2
SOLUTION (Maximized): 237.6587 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
f21	0.0000	0.0003	f22	0.0000	0.0004
f23	1,000.0000	0.0003	f24	200.0000	0.0003
f25	200.0000	0.0073	f26	0.0000	0.0044
f27	1,000.0000	0.0059	f28	0.0000	0.0044
f29	0.0000	0.0026	f30	0.0000	0.0036
f31	0.0000	0.0059	f32	0.0000	0.0036
f33	0.0000	0.0048	f34	0.0000	0.0036
f35	0.0000	0.0022	f36	1,200.0000	0.0030
f37	0.0000	0.0022	f38	0.0000	0.0013
f39	0.0000	0.0018	f40	0.0000	0.0030

File: Output5 10/03/90 09:44:12 Page 1-3
SOLUTION (Maximized): 237.6587 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
f41	0.0000	0.0018	f42	0.0000	0.0024
f43	0.0000	0.0022	f44	0.0000	0.0013
f45	1,200.0000	0.0018	f46	0.0000	0.0013
f47	0.0000	0.0008	f48	0.0000	0.0011
f49	0.0000	0.0018	f50	0.0000	0.0011
f51	0.0000	0.0015	f52	0.0000	0.0073
f53	0.0000	0.0044	f54	1,200.0000	0.0059
f55	0.0000	0.0044	f56	0.0000	0.0026
f57	0.0000	0.0036	f58	0.0000	0.0059
f59	0.0000	0.0036	f60	0.0000	0.0048

File: Output5 10/03/90 09:44:12 Page 1-4

SOLUTION (Maximized): 237.6587 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
I f61	2,200.0000	0.0720	f62	0.0000	0.0432
f63	0.0000	0.0588	d24	0.0000	0.0000
d25	0.0000	0.0000	d26	0.0000	0.0000
d27	0.0000	0.0000	d28	0.0000	0.0000
d29	0.0000	0.0000	d30	0.0000	0.0000
I d31	1,000.0000	0.0000	d32	0.0000	0.0000
d33	0.0000	0.0000	d34	0.0000	0.0000
d35	0.0000	0.0000	d36	0.0000	0.0000
d37	0.0000	0.0000	d38	0.0000	0.0000
d39	0.0000	0.0000	d40	0.0000	0.0000

File: Output5 10/03/90 09:44:12 Page 1-5

SOLUTION (Maximized): 237.6587 Investment Strategy Model 1

Variable	Activity	Cost	Variable	Activity	Cost
d41	0.0000	0.0000	d42	0.0000	0.0000
d43	0.0000	0.0000	d44	0.0000	0.0000
d45	0.0000	0.0000	d46	0.0000	0.0000
d47	0.0000	0.0000	d48	0.0000	0.0000
d49	0.0000	0.0000	d50	0.0000	0.0000

CONSTRAINTS: Investment Strategy Model 1

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 24	1,200.0000	<	1,200.0000	Arc 25	1,200.0000	<	1,200.0000
Arc 26	1,200.0000	<	1,200.0000	Arc 27	1,200.0000	<	1,200.0000
Arc 28	1,200.0000	<	1,200.0000	Arc 29	1,200.0000	<	1,200.0000
Arc 30	1,200.0000	<	1,200.0000	Arc 31	1,200.0000	<	1,200.0000
Arc 32	0.0000	<	1,200.0000	Arc 33	1,200.0000	<	1,200.0000
Arc 34	1,200.0000	<	4,800.0000	Arc 35	1,200.0000	<	4,800.0000
Arc 36	1,200.0000	<	4,800.0000	Arc 37	0.0000	<	4,800.0000
Arc 38	2,400.0000	<	4,800.0000	Arc 39	4,800.0000	<	4,800.0000
Arc 40	0.0000	<	4,800.0000	Arc 41	1,200.0000	<	4,800.0000
Arc 42	4,800.0000	<	4,800.0000	Arc 43	0.0000	<	4,800.0000

CONSTRAINTS: Investment Strategy Model 1

Constraint	Activity		RHS	Constraint	Activity		RHS
Arc 44	1,200.0000	<	4,800.0000	Arc 45	4,800.0000	<	4,800.0000
Arc 46	1,000.0000	<	4,800.0000	Arc 47	4,800.0000	<	4,800.0000
Arc 48	4,800.0000	<	4,800.0000	Arc 49	1,000.0000	<	4,800.0000
Arc 50	4,800.0000	<	4,800.0000	Budget	100000.0000	<	100000.0000

Total Error: 0.000000

A.9.6 Investment Strategy Model 3

```
MODEL:
MAX= R1 * F1 + R2 * F2 + R3 * F3 + R4 * F4 + R5 * F5 +
     R6 * F6 + R7 * F7 + R8 * F8 + R9 * F9 + R10 * F10 +
     R11 * F11 + R12 * F12 + R13 * F13 + R14 * F14 + R15 * F15 +
     R16 * F16 + R17 * F17 + R18 * F18 + R19 * F19 + R20 * F20 +
     R21 * F21 + R22 * F22 + R23 * F23 + R24 * F24 + R25 * F25 +
     R26 * F26 + R27 * F27 + R28 * F28 + R29 * F29 + R30 * F30 +
     R31 * F31 + R32 * F32 + R33 * F33 + R34 * F34 + R35 * F35 +
     R36 * F36 + R37 * F37 + R38 * F38 + R39 * F39 + R40 * F40 +
     R41 * F41 + R42 * F42 + R43 * F43 + R44 * F44 + R45 * F45 +
     R46 * F46 + R47 * F47 + R48 * F48 + R49 * F49 + R50 * F50 +
     R51 * F51 + R52 * F52 + R53 * F53 + R54 * F54 + R55 * F55 +
     R56 * F56 + R57 * F57 + R58 * F58 + R59 * F59 + R60 * F60 +
     R61 * F61 + R62 * F62 + R63 * F63 ;
R1 = ( 0.5 + .1 * X4 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
     ( 1 + .1 * X19 ) * ( 1 + .1 * X20 ) * ( 1 + .1 * X24 ) *
     ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R2 = ( 0.5 + .1 * X4 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
     ( 0.3 + .1 * X18 ) * ( 1 + .1 * X20 ) * ( 1 + .1 * X24 ) *
     ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R3 = ( 0.5 + .1 * X4 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
     ( 0.7 + .1 * X17 ) * ( 1 + .1 * X20 ) * ( 1 + .1 * X24 ) *
     ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R4 = ( 0.3 + .1 * X2 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
     ( 1 + .1 * X19 ) * ( 1 + .1 * X21 ) * ( 0.6 + .1 * X25 ) *
     ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R5 = ( 0.3 + .1 * X2 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
     ( 0.3 + .1 * X18 ) * ( 1 + .1 * X21 ) * ( 0.6 + .1 * X25 ) *
     ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R6 = ( 0.3 + .1 * X2 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
     ( 0.7 + .1 * X17 ) * ( 1 + .1 * X21 ) * ( 0.6 + .1 * X25 ) *
     ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R7 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 1 + .1 * X6 ) *
     ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 1 + .1 * X19 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
     ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) * ( 0.6 + .1 * X45 ) *
     ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R8 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 1 + .1 * X6 ) *
     ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 0.3 + .1 * X18 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
     ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) * ( 0.6 + .1 * X46 ) *
     ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R9 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 1 + .1 * X6 ) *
     ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 0.7 + .1 * X17 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
     ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) * ( 0.3 + .1 * X47 ) *
```

```
    ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R10 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.3 + .1 * X7 ) *
    ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
    ( 0.8 + .1 * X16 ) * ( 1 + .1 * X19 ) * ( 1 + .1 * X21 ) *
    ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
    ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) * ( 0.6 + .1 * X45 ) *
    ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R11 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.3 + .1 * X7 ) *
    ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
    ( 0.8 + .1 * X16 ) * ( 0.3 + .1 * X18 ) * ( 1 + .1 * X21 ) *
    ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
    ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) * ( 0.6 + .1 * X46 ) *
    ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R12 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.3 + .1 * X7 ) *
    ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
    ( 0.8 + .1 * X16 ) * ( 0.7 + .1 * X17 ) * ( 1 + .1 * X21 ) *
    ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
    ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) * ( 0.3 + .1 * X47 ) *
    ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R13 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.7 + .1 * X8 ) *
    ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
    ( 0.8 + .1 * X16 ) * ( 1 + .1 * X19 ) * ( 1 + .1 * X21 ) *
    ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
    ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) * ( 0.6 + .1 * X45 ) *
    ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R14 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.7 + .1 * X8 ) *
    ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
    ( 0.8 + .1 * X16 ) * ( 0.3 + .1 * X18 ) * ( 1 + .1 * X21 ) *
    ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
    ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) * ( 0.6 + .1 * X46 ) *
    ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R15 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.7 + .1 * X8 ) *
    ( 0.8 + .1 * X10 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
    ( 0.8 + .1 * X16 ) * ( 0.7 + .1 * X17 ) * ( 1 + .1 * X21 ) *
    ( 0.3 + .1 * X26 ) * ( 0.6 + .1 * X32 ) * ( 0.6 + .1 * X37 ) *
    ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) * ( 0.3 + .1 * X47 ) *
    ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R16 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 1 + .1 * X6 ) *
    ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
    ( 0.8 + .1 * X16 ) * ( 1 + .1 * X19 ) * ( 1 + .1 * X21 ) *
    ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
    ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) * ( 0.6 + .1 * X45 ) *
    ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R17 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 1 + .1 * X6 ) *
    ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
    ( 0.8 + .1 * X16 ) * ( 0.3 + .1 * X18 ) * ( 1 + .1 * X21 ) *
    ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
    ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) * ( 0.6 + .1 * X46 ) *
    ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R18 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 1 + .1 * X6 ) *
    ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
```

```
     ( 0.8 + .1 * X16 ) * ( 0.7 + .1 * X17 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
     ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) * ( 0.3 + .1 * X47 ) *
     ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R19 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.3 + .1 * X7 ) *
     ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 1 + .1 * X19 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
     ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) * ( 0.6 + .1 * X45 ) *
     ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R20 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.3 + .1 * X7 ) *
     ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 0.3 + .1 * X18 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
     ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) * ( 0.6 + .1 * X46 ) *
     ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R21 = ( 0.3 + .1 * X2 ) * ( 0 8 + .1 * X5 ) * ( 0.3 + .1 * X7 ) *
     ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 0.7 + .1 * X17 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
     ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) * ( 0.3 + .1 * X47 ) *
     ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R22 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.7 + .1 * X8 ) *
     ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 1 + .1 * X19 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
     ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) * ( 0.6 + .1 * X45 ) *
     ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R23 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.7 + .1 * X8 ) *
     ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 0.3 + .1 * X18 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
     ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) * ( 0.6 + .1 * X46 ) *
     ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R24 = ( 0.3 + .1 * X2 ) * ( 0.8 + .1 * X5 ) * ( 0.7 + .1 * X8 ) *
     ( 1 + .1 * X11 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) *
     ( 0.8 + .1 * X16 ) * ( 0.7 + .1 * X17 ) * ( 1 + .1 * X21 ) *
     ( 0.3 + .1 * X26 ) * ( 0.7 + .1 * X33 ) * ( 1 + .1 * X38 ) *
     ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) * ( 0.3 + .1 * X47 ) *
     ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R25 = ( 0.7 + .1 * X3 ) * ( 1 + .1 * X6 ) * ( 1 + .1 * X11 ) *
     ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
     ( 1 + .1 * X19 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
     ( 1 + .1 * X38 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
     ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R26 = ( 0.7 + .1 * X3 ) * ( 1 + .1 * X6 ) * ( 1 + .1 * X11 ) *
     ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
     ( 0.3 + .1 * X18 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
     ( 1 + .1 * X38 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
     ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R27 = ( 0.7 + .1 * X3 ) * ( 1 + .1 * X6 ) * ( 1 + .1 * X11 ) *
```

```
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
    ( 1 + .1 * X38 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R28 = ( 0.7 + .1 * X3 ) * ( 0.3 + .1 * X7 ) * ( 1 + .1 * X11 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
    ( 1 + .1 * X38 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R29 = ( 0.7 + .1 * X3 ) * ( 0.3 + .1 * X7 ) * ( 1 + .1 * X11 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
    ( 1 + .1 * X38 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R30 = ( 0.7 + .1 * X3 ) * ( 0.3 + .1 * X7 ) * ( 1 + .1 * X11 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
    ( 1 + .1 * X38 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R31 = ( 0.7 + .1 * X3 ) * ( 0.7 + .1 * X8 ) * ( 1 + .1 * X11 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
    ( 1 + .1 * X38 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R32 = ( 0.7 + .1 * X3 ) * ( 0.7 + .1 * X8 ) * ( 1 + .1 * X11 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
    ( 1 + .1 * X38 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R33 = ( 0.7 + .1 * X3 ) * ( 0.7 + .1 * X8 ) * ( 1 + .1 * X11 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X27 ) *
    ( 1 + .1 * X38 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R34 = ( 0.7 + .1 * X3 ) * ( 1 + .1 * X6 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R35 = ( 0.7 + .1 * X3 ) * ( 1 + .1 * X6 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R36 = ( 0.7 + .1 * X3 ) * ( 1 + .1 * X6 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R37 = ( 0.7 + .1 * X3 ) * ( 0.3 + .1 * X7 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
```

```
    ( 1 + .1 * X19 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R38 = ( 0.7 + .1 * X3 ) * ( 0.3 + .1 * X7 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R39 = ( 0.7 + .1 * X3 ) * ( 0.3 + .1 * X7 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R40 = ( 0.7 + .1 * X3 ) * ( 0.7 + .1 * X8 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R41 = ( 0.7 + .1 * X3 ) * ( 0.7 + .1 * X8 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R42 = ( 0.7 + .1 * X3 ) * ( 0.7 + .1 * X8 ) * ( 0.5 + .1 * X9 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X22 ) * ( 1 + .1 * X28 ) *
    ( 1 + .1 * X34 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R43 = ( 1 + .1 * X1 ) * ( 1 + .1 * X6 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
    ( 0.7 + .1 * X35 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R44 = ( 1 + .1 * X1 ) * ( 1 + .1 * X6 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
    ( 0.7 + .1 * X35 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R45 = ( 1 + .1 * X1 ) * ( 1 + .1 * X6 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
    ( 0.7 + .1 * X35 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R46 = ( 1 + .1 * X1 ) * ( 0.3 + .1 * X7 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
    ( 0.7 + .1 * X35 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R47 = ( 1 + .1 * X1 ) * ( 0.3 + .1 * X7 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
```

```
    ( 0.7 + .1 * X35 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R48 = ( 1 + .1 * X1 ) * ( 0.3 + .1 * X7 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
    ( 0.7 + .1 * X35 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R49 = ( 1 + .1 * X1 ) * ( 0.7 + .1 * X8 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
    ( 0.7 + .1 * X35 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R50 = ( 1 + .1 * X1 ) * ( 0.7 + .1 * X8 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
    ( 0.7 + .1 * X35 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R51 = ( 1 + .1 * X1 ) * ( 0.7 + .1 * X8 ) * ( 0.3 + .1 * X12 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X29 ) *
    ( 0.7 + .1 * X35 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R52 = ( 1 + .1 * X1 ) * ( 1 + .1 * X6 ) * ( 0.7 + .1 * X13 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R53 = ( 1 + .1 * X1 ) * ( 1 + .1 * X6 ) * ( 0.7 + .1 * X13 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R54 = ( 1 + .1 * X1 ) * ( 1 + .1 * X6 ) * ( 0.7 + .1 * X13 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.3 + .1 * X39 ) * ( 0.6 + .1 * X42 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R55 = ( 1 + .1 * X1 ) * ( 0.3 + .1 * X7 ) * ( 0.7 + .1 * X13 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R56 = ( 1 + .1 * X1 ) * ( 0.3 + .1 * X7 ) * ( 0.7 + .1 * X13 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R57 = ( 1 + .1 * X1 ) * ( 0.3 + .1 * X7 ) * ( 0.7 + .1 * X13 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.6 + .1 * X40 ) * ( 0.6 + .1 * X43 ) *
```

```
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 [illegible] ) ;
R58 = ( 1 + .1 * X1 ) * ( 0.7 + .1 * X8 ) * ( 0.7 + .1 [illegible] ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R59 = ( 1 + .1 * X1 ) * ( 0.7 + .1 * X8 ) * ( 0.7 + .1 * X13 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R60 = ( 1 + .1 * X1 ) * ( 0.7 + .1 * X8 ) * ( 0.7 + .1 * X13 ) *
    ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X15 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X30 ) *
    ( 1 + .1 * X36 ) * ( 0.7 + .1 * X41 ) * ( 0.3 + .1 * X44 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
R61 = ( 1 + .1 * X1 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
    ( 1 + .1 * X19 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X31 ) *
    ( 0.6 + .1 * X45 ) * ( 0.3 + .1 * X48 ) * ( 1 + .1 * X51 ) ;
R62 = ( 1 + .1 * X1 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
    ( 0.3 + .1 * X18 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X31 ) *
    ( 0.6 + .1 * X46 ) * ( 0.6 + .1 * X49 ) * ( 1 + .1 * X51 ) ;
R63 = ( 1 + .1 * X1 ) * ( 0.5 + .1 * X14 ) * ( 0.8 + .1 * X16 ) *
    ( 0.7 + .1 * X17 ) * ( 1 + .1 * X23 ) * ( 1 + .1 * X31 ) *
    ( 0.3 + .1 * X47 ) * ( 0.7 + .1 * X50 ) * ( 1 + .1 * X51 ) ;
1 + .1 * X1 < 1 ;
0.3 + .1 * X2 < 1 ;
0.7 + .1 * X3 < 1 ;
0.5 + .1 * X4 < 1 ;
0.8 + .1 * X5 < 1 ;
1 + .1 * X6 < 1 ;
0.3 + .1 * X7 < 1 ;
0.7 + .1 * X8 < 1 ;
0.5 + .1 * X9 < 1 ;
0.8 + .1 * X10 < 1 ;
1 + .1 * X11 < 1 ;
0.3 + .1 * X12 < 1 ;
0.7 + .1 * X13 < 1 ;
0.5 + .1 * X14 < 1 ;
0.8 + .1 * X15 < 1 ;
0.8 + .1 * X16 < 1 ;
0.7 + .1 * X17 < 1 ;
0.3 + .1 * X18 < 1 ;
1 + .1 * X19 < 1 ;
1 + .1 * X20 < 1 ;
1 + .1 * X21 < 1 ;
1 + .1 * X22 < 1 ;
1 + .1 * X23 < 1 ;
1 + .1 * X24 < 1 ;
0.6 + .1 * X25 < 1 ;
0.3 + .1 * X26 < 1 ;
```

```
1 + .1 * X27 < 1 ;
1 + .1 * X28 < 1 ;
1 + .1 * X29 < 1 ;
1 + .1 * X30 < 1 ;
1 + .1 * X31 < 1 ;
0.6 + .1 * X32 < 1 ;
0.7 + .1 * X33 < 1 ;
1 + .1 * X34 < 1 ;
0.7 + .1 * X35 < 1 ;
1 + .1 * X36 < 1 ;
0.6 + .1 * X37 < 1 ;
1 + .1 * X38 < 1 ;
0.3 + .1 * X39 < 1 ;
0.6 + .1 * X40 < 1 ;
0.7 + .1 * X41 < 1 ;
0.6 + .1 * X42 < 1 ;
0.6 + .1 * X43 < 1 ;
0.3 + .1 * X44 < 1 ;
0.6 + .1 * X45 < 1 ;
0.6 + .1 * X46 < 1 ;
0.3 + .1 * X47 < 1 ;
0.3 + .1 * X48 < 1 ;
0.6 + .1 * X49 < 1 ;
0.7 + .1 * X50 < 1 ;
1 + .1 * X51 < 1 ;
F1 + F2 + F3 < 1200 ;
F4 + F5 + F6 < 1200 ;
F7 + F8 + F9 + F10 + F11 + F12 + F13 + F14 + F15 +
F16 + F17 + F18 + F19 + F20 + F21 + F22 + F23 + F24 < 1200 ;
F25 + F26 + F27 + F28 + F29 + F30 + F31 + F32 + F33 < 1200 ;
F34 + F35 + F36 + F37 + F38 + F39 + F40 + F41 + F42 < 1200 ;
F43 + F44 + F45 + F46 + F47 + F48 + F49 + F50 + F51 < 1200 ;
F52 + F53 + F54 + F55 + F56 + F57 + F58 + F59 + F60 < 1200 ;
F61 + F62 + F63 < 1200 ;
F7 + F8 + F9 + F10 + F11 + F12 + F13 + F14 + F15 < 1200 ;
F16 + F17 + F18 + F19 + F20 + F21 + F22 + F23 + F24 < 1200 ;
F34 + F35 + F36 + F37 + F38 + F39 + F40 + F41 + F42 < 4800 ;
F43 + F44 + F45 + F46 + F47 + F48 + F49 + F50 + F51 < 4800 ;
F52 + F53 + F54 + F55 + F56 + F57 + F58 + F59 + F60 < 4800 ;
F7 + F8 + F9 + F10 + F11 + F12 + F13 + F14 + F15 < 4800 ;
F16 + F17 + F18 + F19 + F20 + F21 + F22 + F23 + F24 +
F25 + F26 + F27 + F28 + F29 + F30 + F31 + F32 + F33 < 4800 ;
F7 + F8 + F9 + F16 + F17 + F18 + F25   F26 + F27 +
F34 + F35 + F36 + F43 + F44 + F45 + F52 + F53 + F54 < 4800 ;
F10 + F11 + F12 + F19 + F20 + F21 + F28 + F29 + F30 +
F37 + F38 + F39 + F46 + F47 + F48 + F55 + F56 + F57 < 4800 ;
F13 + F14 + F15 + F22 + F23 + F24 + F31 + F32 + F33 +
F40 + F41 + F42 + F49 + F50 + F51 + F58 + F59 + F60 < 4800 ;
F7 + F8 + F9 + F16 + F17 + F18 + F25 + F26 + F27 +
F34 + F35 + F36 + F43 + F44 + F45 + F52 + F53 + F54 < 4800 ;
F10 + F11 + F12 + F19 + F20 + F21 + F28 + F29 + F30 +
```

```
F37 + F38 + F39 + F46 + F47 + F48 + F55 + F56 + F57 < 4800 ;
F13 + F14 + F15 + F22 + F23 + F24 + F31 + F32 + F33 +
F40 + F41 + F42 + F49 + F50 + F51 + F58 + F59 + F60 < 4800 ;
F1 + F4 + F7 + F10 + F13 + F16 + F19 + F22 + F25 +
F28 + F31 + F34 + F37 + F40 + F43 + F46 + F49 + F52 +
F55 + F58 + F61 < 4800 ;
F2 + F5 + F8 + F11 + F14 + F17 + F20 + F23 + F26 +
F29 + F32 + F35 + F38 + F41 + F44 + F47 + F50 + F53 +
F56 + F59 + F62 < 4800 ;
F3 + F6 + F9 + F12 + F15 + F18 + F21 + F24 + F27 +
F30 + F33 + F36 + F39 + F42 + F45 + F48 + F51 + F54 +
F57 + F60 + F63 < 4800 ;
F1 + F4 + F7 + F10 + F13 + F16 + F19 + F22 + F25 +
F28 + F31 + F34 + F37 + F40 + F43 + F46 + F49 + F52 +
F55 + F58 + F61 < 4800 ;
F2 + F5 + F8 + F11 + F14 + F17 + F20 + F23 + F26 +
F29 + F32 + F35 + F38 + F41 + F44 + F47 + F50 + F53 +
F56 + F59 + F62 < 4800 ;
F3 + F6 + F9 + F12 + F15 + F18 + F21 + F24 + F27 +
F30 + F33 + F36 + F39 + F42 + F45 + F48 + F51 + F54 +
F57 + F60 + F63 < 4800 ;
100 * X1 + 100 * X2 + 100 * X3 + 100 * X4 + 100 * X5 + 100 * X6 +
100 * X7 + 100 * X8 + 100 * X9 + 100 * X10 + 100 * X11 +
100 * X12 + 100 * X13 + 100 * X14 + 100 * X15 + 100 * X16 +
100 * X17 + 100 * X18 + 100 * X19 + 100 * X20 + 100 * X21 +
100 * X22 + 100 * X23 + 100 * X24 + 100 * X25 + 100 * X26 +
100 * X27 + 100 * X28 + 100 * X29 + 100 * X30 + 100 * X31 +
100 * X32 + 100 * X33 + 100 * X34 + 100 * X35 + 100 * X36 +
100 * X37 + 100 * X38 + 100 * X39 + 100 * X40 + 100 * X41 +
100 * X42 + 100 * X43 + 100 * X44 + 100 * X45 + 100 * X46 +
100 * X47 + 100 * X48 + 100 * X49 + 100 * X50 + 100 * X51  < 100000 ;

END
SLB X1 0
SLB X2 0
SLB X3 0
SLB X4 0
SLB X5 0
SLB X6 0
SLB X7 0
SLB X8 0
SLB X9 0
SLB X10 0
SLB X11 0
SLB X12 0
SLB X13 0
SLB X14 0
SLB X15 0
SLB X16 0
SLB X17 0
SLB X18 0
```

```
SLB X19 0
SLB X20 0
SLB X21 0
SLB X22 0
SLB X23 0
SLB X24 0
SLB X25 0
SLB X26 0
SLB X27 0
SLB X28 0
SLB X29 0
SLB X30 0
SLB X31 0
SLB X32 0
SLB X33 0
SLB X34 0
SLB X35 0
SLB X36 0
SLB X37 0
SLB X38 0
SLB X39 0
SLB X40 0
SLB X41 0
SLB X42 0
SLB X43 0
SLB X44 0
SLB X45 0
SLB X46 0
SLB X47 0
SLB X48 0
SLB X49 0
SLB X50 0
SLB X51 0
SLB F1 0
SLB F2 0
SLB F3 0
SLB F4 0
SLB F5 0
SLB F6 0
SLB F7 0
SLB F8 0
SLB F9 0
SLB F10 0
SLB F11 0
SLB F12 0
SLB F13 0
SLB F14 0
SLB F15 0
SLB F16 0
SLB F17 0
SLB F18 0
```

```
SLB F19 0
SLB F20 0
SLB F21 0
SLB F22 0
SLB F23 0
SLB F24 0
SLB F25 0
SLB F26 0
SLB F27 0
SLB F28 0
SLB F29 0
SLB F30 0
SLB F31 0
SLB F32 0
SLB F33 0
SLB F34 0
SLB F35 0
SLB F36 0
SLB F37 0
SLB F38 0
SLB F39 0
SLB F40 0
SLB F41 0
SLB F42 0
SLB F43 0
SLB F44 0
SLB F45 0
SLB F46 0
SLB F47 0
SLB F48 0
SLB F49 0
SLB F50 0
SLB F51 0
SLB F52 0
SLB F53 0
SLB F54 0
SLB F55 0
SLB F56 0
SLB F57 0
SLB F58 0
SLB F59 0
SLB F60 0
SLB F61 0
SLB F62 0
SLB F63 0

LEAVE


ODEGENERATE FOR   15 STEPS.  PROBABLY CYCLING.
 SOLUTION STATUS:  OPTIMAL TO TOLERANCES.  DUAL CONDITIONS:  SATISFIED.
```

OBJECTIVE FUNCTION VALUE

1) 4266.039552

VARIABLE	VALUE	REDUCED COST
R1	1.000000	0.000000
F1	67.576792	-1.000000
R2	1.000000	0.000000
F2	67.549448	-1.000000
R3	1.000039	0.000000
F3	67.716996	-1.000039
R4	1.000000	0.000000
F4	67.338006	-1.000000
R5	1.000000	0.000000
F5	67.325673	-1.000000
R6	1.000039	0.000000
F6	67.453597	-1.000039
R7	1.000000	0.000000
F7	66.501984	0.000039
R8	1.000000	0.000000
F8	66.518607	0.000039
R9	1.000039	0.000000
F9	66.569114	0.000000
R10	1.000000	0.000000
F10	66.571720	0.000039
R11	1.000000	0.000000
F11	66.588729	0.000039
R12	1.000039	0.000000
F12	66.638295	0.000000
R13	1.000000	0.000000
F13	66.535607	0.000039
R14	1.000000	0.000000
F14	66.555336	0.000039
R15	1.000039	0.000000
F15	66.597028	0.000000
R16	1.000000	0.000000
F16	66.709600	0.000039
R17	1.000000	0.000000
F17	66.725567	0.000039
R18	1.000039	0.000000
F18	66.777898	0.000000
R19	1.000000	0.000000
F19	66.778936	0.000039
R20	1.000000	0.000000
F20	66.795324	0.000039
R21	1.000039	0.000000
F21	66.846630	0.000000
R22	1.000000	0.000000
F22	66.736052	0.000039
R23	1.000000	0.000000

F23	66.755381	0.000039
R24	1.000039	0.000000
F24	66.798192	0.000000
R25	1.000000	0.000000
F25	67.933595	-1.000000
R26	1.000000	0.000000
F26	67.929844	-1.000000
R27	1.000039	0.000000
F27	68.038059	-1.000039
R28	1.000000	0.000000
F28	67.991537	-1.000000
R29	1.000000	0.000000
F29	67.989196	-1.000000
R30	1.000039	0.000000
F30	68.093971	-1.000039
R31	1.000000	0.000000
F31	67.717494	-1.000000
R32	1.000000	0.000000
F32	67.724753	-1.000000
R33	1.000039	0.000000
F33	67.802089	-1.000039
R34	1.000000	0.000000
F34	68.056924	-1.000000
R35	1.000000	0.000000
F35	68.057351	-1.000000
R36	1.000039	0.000000
F36	68.154312	-1.000039
R37	1.000000	0.000000
F37	68.117646	-1.000000
R38	1.000000	0.000000
F38	68.119247	-1.000000
R39	1.000039	0.000000
F39	68.213341	-1.000039
R40	1.000000	0.000000
F40	67.878572	-1.000000
R41	1.000000	0.000000
F41	67.888361	-1.000000
R42	1.000039	0.000000
F42	67.958854	-1.000039
R43	1.000000	0.000000
F43	67.951886	-1.000000
R44	1.000000	0.000000
F44	67.958260	-1.000000
R45	1.000039	0.000000
F45	68.037702	-1.000039
R46	1.000000	0.000000
F46	68.015664	-1.000000
R47	1.000000	0.000000
F47	68.022943	-1.000000
R48	1.000039	0.000000
F48	68.100177	-1.000039

R49	1.000000	0.000000
F49	67.859505	-1.000000
R50	1.000000	0.000000
F50	67.872963	-1.000000
R51	1.000039	0.000000
F51	67.932520	-1.000039
R52	1.000000	0.000000
F52	68.579090	-1.000000
R53	1.000000	0.000000
F53	68.575384	-1.000000
R54	1.000039	0.000000
F54	68.683034	-1.000039
R55	1.000000	0.000000
F55	68.636854	-1.000000
R56	1.000000	0.000000
F56	68.634575	-1.000000
R57	1.000039	0.000000
F57	68.738742	-1.000039
R58	1.000000	0.000000
F58	68.369804	-1.000000
R59	1.000000	0.000000
F59	68.377106	-1.000000
R60	1.000039	0.000000
F60	68.454033	-1.000039
R61	1.000000	0.000000
F61	69.478808	-1.000000
R62	1.000000	0.000000
F62	69.437605	-1.000000
R63	1.000039	0.000000
F63	69.641868	-1.000039
X4	5.000000	0.000000
X14	5.000000	0.000000
X16	2.000000	0.000000
X19	0.000000	0.000000
X20	0.000000	-20.284588
X24	0.000000	-20.284588
X45	4.000000	0.000000
X48	7.000000	0.000000
X51	0.000000	0.000000
X18	7.000000	0.000000
X46	4.000000	0.000000
X49	4.000000	0.000000
X17	3.000389	0.000000
X47	7.000000	0.000000
X50	3.000000	0.000000
X2	7.000000	0.000000
X21	0.000000	0.000000
X25	4.000000	0.000000
X5	2.000000	0.000000
X6	0.000000	0.000000
X10	2.000000	0.000000

X15	2.000000	0.000000
X26	7.000000	0.000000
X32	4.000000	0.000000
X37	4.000000	0.000000
X39	7.000000	0.000000
X42	4.000000	0.000000
X7	7.000000	0.000000
X40	4.000000	0.000000
X43	4.000000	0.000000
X8	3.000000	0.000000
X41	3.000000	0.000000
X44	7.000000	0.000000
X11	0.000000	0.000000
X33	3.000000	0.000000
X38	0.000000	0.000000
X3	3.000000	0.000000
X22	0.000000	0.000000
X27	0.000000	-61.122849
X9	5.000000	0.000000
X28	0.000000	0.000000
X34	0.000000	0.000000
X1	0.000000	0.000000
X12	7.000000	0.000000
X23	0.000000	0.000000
X29	0.000000	0.000000
X35	3.000000	0.000000
X13	3.000000	0.000000
X30	0.000000	0.000000
X36	0.000000	0.000000
X31	0.000000	-20.856100

ROW	SLACK OR SURPLUS	PRICE
2)	0.000000	67.576792
3)	0.000000	67.549448
4)	0.000000	67.716996
5)	0.000000	67.338006
6)	0.000000	67.325673
7)	0.000000	67.453597
8)	0.000000	66.501984
9)	0.000000	66.518607
10)	0.000000	66.569114
11)	0.000000	66.571720
12)	0.000000	66.588729
13)	0.000000	66.638295
14)	0.000000	66.535607
15)	0.000000	66.555336
16)	0.000000	66.597028
17)	0.000000	66.709600
18)	0.000000	66.725567
19)	0.000000	66.777898
20)	0.000000	66.778936

21)	0.000000	66.795324
22)	0.000000	66.846630
23)	0.000000	66.736052
24)	0.000000	66.755381
25)	0.000000	66.798192
26)	0.000000	67.933595
27)	0.000000	67.929844
28)	0.000000	68.038059
29)	0.000000	67.991537
30)	0.000000	67.989196
31)	0.000000	68.093971
32)	0.000000	67.717494
33)	0.000000	67.724753
34)	0.000000	67.802089
35)	0.000000	68.056924
36)	0.000000	68.057351
37)	0.000000	68.154312
38)	0.000000	68.117646
39)	0.000000	68.119247
40)	0.000000	68.213341
41)	0.000000	67.878572
42)	0.000000	67.888361
43)	0.000000	67.958854
44)	0.000000	67.951886
45)	0.000000	67.958260
46)	0.000000	68.037702
47)	0.000000	68.015664
48)	0.000000	68.022943
49)	0.000000	68.100177
50)	0.000000	67.859505
51)	0.000000	67.872963
52)	0.000000	67.932520
53)	0.000000	68.579090
54)	0.000000	68.575384
55)	0.000000	68.683034
56)	0.000000	68.636854
57)	0.000000	68.634575
58)	0.000000	68.738742
59)	0.000000	68.369804
60)	0.000000	68.377106
61)	0.000000	68.454033
62)	0.000000	69.478808
63)	0.000000	69.437605
64)	0.000000	69.641868
65)	0.000000	1437.377145
66)	0.000000	1402.135447
67)	0.000000	1223.681012
68)	0.000000	202.845876
69)	0.000000	1200.015546
70)	0.000000	1217.774030
71)	0.000000	1218.909360

72)	0.000000	1215.829450
73)	0.000000	612.452574
74)	0.000000	599.084197
75)	0.000000	1212.159864
76)	0.000000	611.759558
77)	0.000000	617.056647
78)	0.000000	4266.039670
79)	0.000000	3652.512763
80)	0.000000	4266.039670
81)	-0.000039	1423.246439
82)	0.000000	1421.401651
83)	0.000000	1421.336038
84)	0.000000	0.000000
85)	0.000000	1402.135447
86)	0.000000	1223.681012
87)	0.000000	1437.377145
88)	0.000000	0.000000
89)	0.000000	202.119901
90)	0.000000	1200.015546
91)	0.000000	0.000000
92)	0.000000	612.452574
93)	0.000000	611.759558
94)	0.000000	617.056647
95)	0.000000	0.000000
96)	0.000000	599.084197
97)	0.000000	600.931388
98)	0.000000	612.452574
99)	0.000000	611.759558
100)	0.000000	617.056647
101)	0.000000	599.084197
102)	0.000000	1212.159864
103)	0.000000	1217.774030
104)	0.000000	1218.909360
105)	0.000000	1215.829450
106)	0.000000	1217.774030
107)	0.000000	1218.909360
108)	0.000000	1215.829450
109)	0.000000	1421.336038
110)	0.000000	1421.401651
111)	0.000000	1423.301828
112)	0.000000	1421.336038
113)	0.000000	1421.401651
114)	0.000000	1423.301828
115)	0.000000	4266.039670
116)	997.156763	0.000000
117)	997.882724	0.000000
118)	0.000000	1.000039
119)	588.779461	0.000000
120)	587.555390	0.000000
121)	588.248382	0.000000
122)	582.951378	0.000000

123)	[illegible]1.441719	0.000000
124)	[illegible]0.923579	0.000000
125)	[illegible]9.076421	0.000000
126)	4187.555390	0.000000
127)	4188.248382	0.000000
128)	4182.951378	0.000000
129)	4200.923579	0.000000
130)	3587.855882	0.000000
131)	3582.241788	0.000000
132)	3581.106473	0.000000
133)	3584.186350	0.000000
134)	3582.241788	0.000000
135)	3581.106473	0.000000
136)	3584.186350	0.000000
137)	3378.663924	0.000000
138)	3378.598346	0.000000
139)	3376.753547	0.000000
140)	3378.663924	0.000000
141)	3378.598346	0.000000
142)	3376.753547	0.000000
143)	85399.961588	0.000000

A.9.7 Output From Maxflo

```
   Original Network

NORMAL STATISTICS
 Mean:   623.520
 Std. Dev.:  1529.28
 Confidence intvl. (+-):  29.9739

 Reliability:
 0.151990

     Network After Investment Strategy 1

NORMAL STATISTICS
 Mean:   698.020
 Std. Dev.:  1678.92
 Confidence intvl. (+-):  32.9068

 Reliability:
 0.151990

     Network After Investment Strategy 3

NORMAL STATISTICS
 Mean:   9600.00
 Std. Dev.: 0.0
 Confidence intvl. (+-):  0.0

 Reliability:
 1.00000
```

Appendix B. *Formula Version 2.0 User's Manual*

This manual explains how to use Prolog program, FORMULA Version 2.0. This program consists of two files:

* FORMULA2.ARI - Contains the main computer program.

* WINDOWS2.ARI - Contains window dialog boxes.

B.1 Required Equipment

Currently, FORMULA requires ARITY/PROLOG interpreter program to run. The interpreter and FORMULA can be installed on an IBM-XT/AT compatible microcomputer. The computer with at least 512 kilobytes of random access memory and 20 megabyte hard disk is desired.

B.2 Running the FORMULA

Assuming both ARITY/PROLOG interpreter and FORMULA are installed on your computer, start the interpreter by typing 'API'. When '?-' prompt appears, consult your program by typing ' consult('formula2.ari'). ' After the program has been consulted correctly, type in 'go.' to start the program FORMULA. The program proceeds through the following steps:

1) First, it displays an introductory screen displaying what this program can do. (Just hit any key to go on.)

2) Next, it asks for the name of input data file which contains the description of network to be analyzed. (Type in the exact name of input file and hit return.)

3) After the input file name has been typed in, the program displays a menu window from which you can choose to generate the specific output (Select number 1, 2, 3, 4, 5, 6, 7, or 8):

1. Find all paths and calculate path reliabilities.
2. Generate the Maximum Flow Formulation.
3. Generate the Lower Bound Formulation.
4. Generate the Upper Bound Formulation.
5. Generate the Investment Strategy Model 1.

6. Generate the Investment Strategy Model 2.

7. Generate the Investment Strategy Model 3.

8. Exit.

4) It then asks where to send the output. If you just want to screen the output, choose 1; otherwise, choose 2 to save the output in a file. The output filename is automatically generated by the program. When the user requests the output to be sent to a file, the output of paths and reliabilities is sent to 'output1.lp', maximum flow to 'output2.lp', lower bound to 'output3.lp', upper bound to 'output4.lp', investment strategy model 1 to 'output5.lp', investment strategy model 2 to 'output6.lp', and investment strategy model 3 to 'output7.nlp'. (Select 1 or 2.)

5) After the output has been generated, the program asks if you want to run the program again. If you are interested in getting other output, type in 'y'; otherwise type in 'n' which exits the program. (Type in y or n.)

B.3 Input

In preparing the input data, you have to follow the following procedures:

1) If the network contains stochastic and/or capacitated nodes, convert the nodes to a dummy arc joined by two nodes. The dummy arc represents the stochastic and/or capacitated node.

2) Introduce an artificial single source and single sink. The source and sink must be named s and t, respectively. Connect all source nodes in the network to s, and all sink nodes to t.

3) Draw a revised network, and assign arc numbers. It has been found very helpful if you number the dummy arcs representing the node with the same node number. Then number the remaining arcs starting with one number higher than the number of nodes in the network. Thus, for example, if you have 20 nodes in the original network, number the remaining arcs starting with 21.

4) Using a text editor (Arity/Prolog comes with its own editor), prepare the input data by typing in the description of revised network that is to be analyzed. The input consists of eight data sets. description of an arc relationship with respect to one another, the survival probability of each arc, the capacity of each arc, the ost of improving each arc by one unit of capacity, the predetermined amount of capacity increase in each component, total budget available for capacity investment, the cost of improving the reliability of each arc by 0.1, and the total budget available for reliability investment.. The input are described as *facts* in Prolog term as follows:

Input	Description
`arc(Arc1,Arc2).`	`Arc1 is the parent of Arc2 (Arc1 precedes Arc2).`
`prob(A,Pb).`	`The survival probability of arc A is Pb.`
`cap(A,Cp).`	`The capacity of arc A is Cp. The unlimited (infinite) capacity is denoted by '*'.`
`cost(A,Cs).`	`The cost of increasing one unit of capacity in arc A is Cs.`
`invest(A,Am).`	`The predetermined amount of capacity increase for arc A is Am.`
`budget(Bc).`	`Total budget available for investment in capacity is Bc.`
`rcost(A,Rs).`	`The cost of increasing one unit of reliability in arc A is Rs.`
`rbudget(Br).`	`Total budget available for investment in reliability is Br.`

When defining arc relationships, you must define the relationship of *s* and *t* with respect to other arcs. So '`arc(s,2).`' defines that the arc 2 is incidence to node `s`; that is, the arc 2 leaves the node `s`. In a likely manner, '`arc(22,t).`' defines that the arc 22 arrives at node `t`. In addition to specifying the probability of each arc, the probability of `s` and `t` must be defined as 1. Thus, in the probability description, make sure you include '`prob(s,1)`' and '`prob(t,1)`'.

5) Depending on your need, some of the input data may be omitted. Refer to the table below to see exactly which data set are required to get the desired output. For example, if you are only interested in finding paths and their reliabilities, all you need is arc relationships and survival probabilities.

For this output:	You must have the following data:
Paths and Reliabilities	`arc(X,Y).`
	`prob(X,Y).`

```
|Maximum Flow,           | arc(X,Y).                         |
|Lower Bound, or         | prob(X,Y).                        |
|Upper Bound             | cap(X,Y).                         |
---------------------------------------------------------------
|Improvement Strategy 1  | arc(X,Y).                         |
|                        | prob(X,Y).                        |
|                        | cap(X,Y).                         |
|                        | cost(X,Y).                        |
|                        | budget(X,Y).                      |
---------------------------------------------------------------
|Improvement Strategy 2  | arc(X,Y).                         |
|                        | prob(X,Y).                        |
|                        | cap(X,Y).                         |
|                        | cost(X,Y).                        |
|                        | budget(X,Y).                      |
|                        | invest(X,Y).                      |
---------------------------------------------------------------
|Improvement Strategy 3  | arc(X,Y).                         |
|                        | prob(X,Y).                        |
|                        | cap(X,Y).                         |
|                        | rcost(X,Y).                       |
|                        | rbudget(X,Y).                     |
---------------------------------------------------------------
```

6) When using Arity/Prolog, it is customary to name the file with '.ari' extension. So name your input file with '.ari' extension.

B.4 Example

To illustrate how to prepare the input data, consider a network shown in Figure 17. This network contains 4 nodes and 3 arcs. It has multiple sources, 1 and 2, and a single sink, 4.

The cost of increasing one unit of capacity is as follows: node 3 = 10, $arc_{1,3}$ = 20, $arc_{2,3}$ = 30, and $arc_{3,4}$ = 40. The predetermined amount of capacity increase is 5 units for all components, and the total budget available for investment is $1000. The cost of increasing reliability 0.1 is the same as increasing capacity by 1, and the reliability budget is the same as the capacity budget. The revised network is shown in Figure 18.

Since node 1 and 4 are not stochastic and not capacitated, they do not need to be represented as arcs. A dummy arc representing a stochastic or capacitated node is assigned the same number as its node number. In numbering arcs, the number 1 and 4 are not used, since 1 and 4 are not represented as arcs. The remaining arcs are numbered starting with 5. Now, referring to the revised network shown in Figure 18, the input data can be prepared as follows:

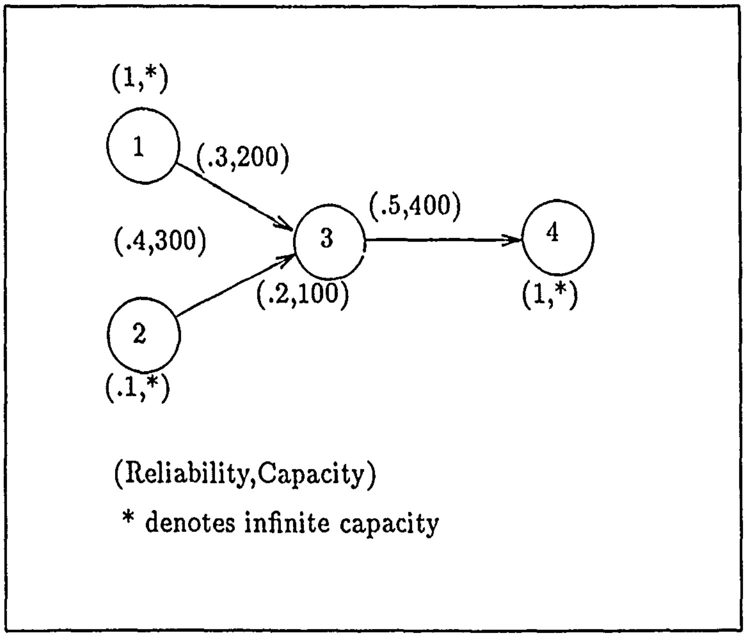


Figure 17. Sample Network

```
% Arc Relationship
   arc(s,5).
   arc(s,6).
   arc(2,8).
   arc(3,9).
   arc(5,7).
   arc(6,2).
   arc(7,3).
   arc(8,3).
   arc(9,10).
   arc(10,t).

% Survival Probabilities
   prob(s,1).
   prob(2,0.1).
   prob(3,0.2).
   prob(5,1).
   prob(6,1).
   prob(7,0.3).
   prob(8,0.4).
   prob(9,0.5).
   prob(10,1).
   prob(t,1).
```

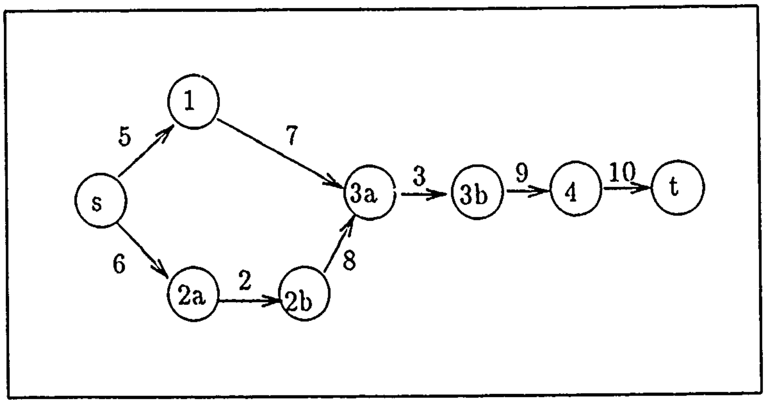


Figure 18. Revised Network

```
% Capacity
   cap(3,100).
   cap(5,*).
   cap(6,*).
   cap(7,200).
   cap(8,300).
   cap(9,400).
   cap(10,*).

% Cost of Increasing One unit of Capacity
   cost(3,10).
   cost(7,20).
   cost(8,30).
   cost(9,40).

% Predetermined Amount of Capacity Increase
   invest(_,5).

% Capacity Budget Available
   budget(1000).

% Cost of Increasing Reliability by 0.1'
   rcost(3,10).
   rcost(7,20).
   rcost(8,30).
   rcost(9,40).

% Reliability Budget Available
   rbudget(1000).

% --- end--- %
```

Any line that starts with a % sign is a comment line, and it is ignored by the interpreter. The underscore (_) in '`invest(_,5).`' denotes *all components.* Thus, '`invest(_,5).`' denotes the predetermined amount of capacity increase for all components is 5 units.

B.5 Output

An example of output1.lp, containing paths and path reliabilities, are shown below:

```
*************************************************
* Following is a list of all paths from "s" to "t" *
* of the network described in the input data file. *
*************************************************

  Path1: s 5 7 3 9 10 t
  Reliability:  0.003

  Path2: s 6 2 8 3 9 10 t
  Reliability:  0.004

* ----- end ----- *
```

The rest of the outputs, containing mathematical programming models, are in the same format as the input format of LP/MIP-83. Each output consists of ..Title, ..Objective Maximize, and ..Constraints section. An example of output2.lp, containing the maximum flow formulation, is shown below:

```
..Title

Maximum Flow Formulation 2

..Objective Maximize

f1 + f2

..Constraints

Arc 3: f1 + f2 <= 100

Arc 7: f1 <= 200

Arc 8: f2 <= 300

Arc 9: f1 + f2 <= 400
```

```
* ----- end ----- *
```

B.6 LP/MIP-83 Commands

All models, except Investment Strategy Models 2 and 3, can be solved using either LP83 or MIP83. The Investment Strategy Model 2 can only be solved by MIP83, because it requires integer solutions. The following are some commands to run LP/MIP-83:

a) c :> lp83 a:output2

Find all solutions to the linear programming model stored in 'output2.lp' in 'a' drive, and display the solutions on the screen.

b) c :> lp83 a:output2 output a:list

Same as a) above, except send the solutions to the output file named 'list.prn', in 'a' drive. The extension, '.prn' is automatically added.

c) c :> lp83 a:output2 output a:list alternate 1

Same as b) above, except find only one solution.

d) c :> mip83 a:output6

Find all solutions to the mixed integer programming model stored in 'output6.lp' in 'a' drive. and display the solutions on the screen.

B.7 GINO Commands

Investment Strategy Model 3 must be solved using *GINO*. *GINO* can be run on many different types of computer systems. The following example is for a computer running the MS-DOS operating system:

c :> GINO	(this command starts the *GINO* program)
:retr output7.nlp	(this command loads the file created by *Formula*)
:go	(this command tells *GINO* to solve the model)

:quit (this command exits *GINO*)

The user should consult a *GINO* user's manual for further details.

B.8 Helpful Comments

1) The program, FORMULA, has been tested and successfully generated formulations for the network containing 70 nodes, 112 arcs, and 2198 paths. Since the capacity of LP/MIP-83 is approximately 1200 variables (paths), any problem bigger than the capacity of LP/MIP-83, must be solved using other mathematical programming packages, such as MINOS or SAS. Of course preparing an input file for these packages will be different from that of LP/MIP-83.

2) If you want to stop the execution (or if the computer is hung up), press 'control' and 'break' simultaneously. When '?-' appears, type 'clear_windows.' and/or 'exit_popup.'. It will get you to the main window of Arity/Prolog. Before starting the program again, you must erase the datafile by typing 'Restore.' Otherwise, you will get erroneous output.

3) Any questions about the Arity/Prolog interpreter or general questions about the Arity/Prolog, refer to (3). Any questions on LP/MIP-83, refer to (21). For *GINO* questions, see (15).

Appendix C. *Formula Version 2.0 Source Code*

```
/*============================================================*/
/*                                                            */
/*                  FORMULA   Version 2.0                     */
/*                                                            */
/*============================================================*/
/*                                                            */
/* This program does the following seven tasks:               */
/*                                                            */
/*  1) Finds all paths in the network from source (s) to      */
/*     sink (t) and calculates all path reliabilities.        */
/*  2) Generates the formulation of the Maximum Flow through  */
/*     the network.                                           */
/*  3) Generates the formulation of the Lower Bound of the    */
/*     Expected Maximum Flow.                                 */
/*  4) Generates the formulation of the Upper Bound of the    */
/*     Expected Maximum Flow.                                 */
/*  5) Generates the formulation of the investment strategy   */
/*     model 1.                                               */
/*  6) Generates the formulation of the investment strategy   */
/*     model 2.                                               */
/*  7) Generates the formulation of the investment strategy   */
/*     model 3.                                               */
/*                                                            */
/* The six mathematical programming models (2 to 7) are       */
/* developed based on arc-path incidence matrix built from    */
/* the description of the network in the input file.          */
/* The network described in the input file must contain a     */
/* single source node, named 's', and sink node, named 't'.   */
/* All capacitated or stochastic nodes must be represented    */
/* as a dummy arc with two nodes. Refer to FORMULA user's     */
/* manual for details on how to prepare the input file.       */
/*                                                            */
/* The formulations, 2 thru 6, generated from this program    */
/* are in the same format as the input data file of LP/MIP 83 */
/* mathematical programming package.  The formulation 7 is in */
/* the same format as the input data file of GINO mathematical*/
/* programming package.  Thus the ouputs 2 thru 6 can be used */
/* as an input to LP/MIP 83 and the output 7 can be used as   */
/* an input to GINO for further analysis.                     */
/*                                                            */
/*------------------------------------------------------------*/
/*                                                            */
/*   DATE: 1 October, 1990                                    */
/*   FILENAME: FORMULA2.ARI                                   */
/*                                                            */
```

```
/* This program was written in Prolog language using the      */
/* Arity/Prolog Version 5.0.                                   */
/*                                                             */
/*=============================================================*/

/* Start the program by typing 'go.' */

 go :-
     fileerrors(_,off),      % Turn off system file error message.
     windows,                % Call windows to display the
                             % introduction screen.
     open_input_datafile,    % Get the name of input file and
                             % consult it.
     start_program.
 go.

 start_program :-
     get_selection_number(Selection),
                              % Ask user what need to be done.
   ( Selection = 8,           % If '8'is selected, exit.
     clear_windows,!
   ;                          % Otherwise,
     nl,                      % Where should output be sent ?
     get_where_to_send_output(Where),
     execute_request(Selection,Where)   % Execute the request.
   ).


 execute_request(Selection,Where) :-
    ( Selection = 1,
      search_paths(Where),!
    ;
      Selection >= 2,
      Selection =< 4,
      performance_formulations(Selection,Where)
    ;
      Selection >= 5,
      Selection =< 6,
      lp_investment_formulations(Selection,Where)
    ;
      nlp_investment_formulation(Selection,Where)
    ),
    get_run_again_reply(Reply),     % Run the program again ?
    cls,                            % Clear screen.
    ( Where = 1,                    % If the output was sent to
      true, !                       % the screen, do nothing.
    ;                               % Otherwise,
      exit_popup                    % delete popup window 'done'.
    ),
      want_more(Reply).
```

```
want_more(Reply) :-
   ( Reply = 121,              % Reply is 'y' (ASCII 121),
     removeallh(matrix),       % delete hash table 'matrix', and
     start_program,!           % run program again.
   ;
    clear_windows,             % Otherwise, exit.
     nl
   ),!.


/*********************************************************************/
/* 'search_path' initiates search to find all paths in the network */
/* and calculates path reliabilities.                              */
/*********************************************************************/

search_paths(Where) :-
   ctr_set(1,1),          % Initialize counter one to 1
                          % to keep track of path number.
  ( Where = 1,            % When Where = 1, output is displayed
                          % on the monitor screen.
   continue_find_paths

  ;                       % 'Executing' message is displayed on
                          % the screen while the output is being
   executing_message,  % sent to the 'output1.lp' file.

   stdout('output1.lp',continue_find_paths),

   exit_popup,            % Delete popup window 'executing'.
   done_message(1)        % Display 'done' message.
  ).


continue_find_paths :-
   nl,
   write('*********************************************************'),
   nl,
   write('*  Following is a list of all paths from "s" to "t"     *'),
   nl,
   write('*  of the network described in the input data file.    *'),
   nl,
   write('*********************************************************'),
   nl,nl,nl,
   find_paths(s,t),
   nl,nl,
   write('* ------ end ------ *'),
   nl,nl,nl,nl.

/*-----------------------------------------------------------------*/

find_paths(Start,Goal) :-        % Find all paths using depth-first
```

```
                                % search method.
   depth_first([Start],Goal,Path),
   find_reliability(Path,Rel),  % Calculates the reliability of
                                % a path.
   ctr_inc(1,Pnbr),             % Get current path number and
                                % increment counter one by 1.
   display_outputs(Path,Pnbr,Rel),
   fail.                        % Go on to find next path until
                                % there isn't any to be searched.
find_paths(_,_).


depth_first(Path,Goal,Path) :-  % Path is found if it satisfies
   satisfies(Path,Goal).        % Goal.

depth_first([X|Rest],Goal,Path) :-
   arc(X,Y),                    % Get next arc.
   not member(Y,[X|Rest]),      % Prevents cycles.
   depth_first([Y,X|Rest],Goal,Path).  % Recursive call.

satisfies([Goal|_],Goal).       % A path is found if the head of
                                % a list describing Path matches
                                % with Goal (t).


member(X,[X|Tail]).
member(X,[Head|Tail]) :-
   member(X,Tail).


find_reliability([],1) :- !.    % Reliability of an empty list is 1.
find_reliability([Arc|Rest],Rel) :-
   find_reliability(Rest,RelRest),  % Calculate Rel recursively.
   prob(Arc,Pb),                    % Get survival probability of Arc.
   Rel is Pb * RelRest.

/*----------------------------------------------------------*/

display_outputs(Path,Pnbr,Rel) :-
   print_path(Path,Pnbr),             % Print path and
   print_reliability(Rel).            % reliability.


print_path(Path,Pnbr) :-
   write('  Path '),
   write(Pnbr),
   write(': '),
   write_reverse(Path).


print_reliability(Rel) :-
   nl,
```

```
   write('  Reliability: '),
   write(Rel), nl, nl.


write_reverse([]) :- !.            % Prints path from 's' to 't'.
write_reverse([Arc|Rest]) :-
   write_reverse(Rest),
   write(Arc), write(' ').


/*********************************************************************/
/* 'performance_formulations' generates the formulations of maximum */
/* flow, lower bound of expected maximum flow, and upper bound of   */
/* expected  maximum flow.                                          */
/*********************************************************************/


performance_formulations(Selection,Where) :-

  ( Where = 1,      % Display output on the screen.
    find_formulations(Selection)
  ;
    executing_message,              % Display output sending message.
    ( Selection = 2,
      stdout('output2.lp',find_formulations(Selection))
    ;
      Selection = 3,
      stdout('output3.lp',find_formulations(Selection))
    ;
      stdout('output4.lp',find_formulations(Selection))
    ),
    exit_popup,                     % Delete popup window 'executing'.
    done_message(Selection)         % Display 'done' message.
  ).

find_formulations(Selection) :-
    title(Selection),
    objective(Selection),
    constraints(Selection),
    nl,nl,
    write('* ------ end ------ *'),
    nl,nl,nl,nl.


/*-----------------------------------------------------------------*/

title(Selection) :-                 % Print title of the formulation.
   write(' ..Title'),
   nl,nl,
   ( Selection = 2,
     write(' Maximum Flow Formulation')
```

```
   ;
     Selection = 3,
     write(' Lower Bound Formulation')
   ;
     write(' Upper Bound Formulation')
   ).
/*----------------------------------------------------------------*/

objective(Selection) :-            % Get objective function
   nl,nl,
   write(' ..Objective Maximize'),
   nl,nl,
   ctr_set(1,1),  % Counter one generates path number.
   ctr_set(3,1),  % Initialize counter three to 1
                        % to keep track of how many terms
                        % are printed in a line in the objective
                        % function.
   find_objective(s,t,Selection).


find_objective(Start,Goal,Selection) :-
   depth_first([Start],Goal,Path),
   find_reliability(Path,Rel),
   ctr_inc(1,Pnbr),                % Get current path number and
                                   % increment the counter one by one.
   print_objective(Pnbr,Rel,Selection),
   make_arc_path_matrix(Path,Pnbr), % Make arc-path incidence
                                    % matrix.
   fail.
find_objective(_,_,_).


print_objective(Pnbr,Rel,Selection) :-
   tab(1),
   ( Pnbr = 1
   ;
     write('+ ')
   ),
   ( ctr_inc(3,Value),    % Get current variable number and
                          % increment the counter three by 1.
     Value > 4,
     Mod_Value is Value mod 4,   % If the remainder of Value
     Mod_Value = 1,              % divided by 4 is 1, then skip
     nl,                         % to next line.
     write(' ')
   ;
     true
   ),
   ( Selection = 3,             % If finding lower bound (Sel = 3),
     write(Rel)                 % print reliability.
   ;
```

```
     true                          % Otherwise, do nothing.
   ),
   write(' f'),                    % Print path flow variable.
   write(Pnbr), !.

/*-----------------------------------------------------------------*/

make_arc_path_matrix([],_).
make_arc_path_matrix(Arc_List,Pnbr) :-   % Seperate the elements,
                                         % arcs, in the path and
                                         % store each arc with
                                         % associated path number
                                         % to form arc-path incidence
   get_arc(Arc_List,Arc,Rem_List),       % matrix.
   cap(Arc,Capacity),
   ( number(Capacity),
     recordh(matrix,Arc,arc_path_matrix(Arc,Pnbr))
   ;
     true
   ),
   make_arc_path_matrix(Rem_List,Pnbr), !.


get_arc([Head|[H|Rest]],Head,Rem_List) :-
   Head \= 's',                       % Ignore 's' and 't'
   Head \= 't',
   ( Rest = [],
     Rem_List = []
   ;
     Rem_List = [H|Rest]
   ), !.

get_arc([Head|Rest],Arc,Rem_List) :-
   get_arc(Rest,Arc,Rem_List).         % Get one arc at a time
                                       % that is in the path.


/*-----------------------------------------------------------------*/

constraints(Selection) :-              % Get constraints function.
    nl, nl,
    write(' ..Constraints'),
    asserta(init_string($$)),          % Initialize to empty string.
    find_all_arc_lists(Arc_List),
    sort(Arc_List,Sor),                % Sort Arc_List in ascending order.
    make_arc_array(Sor),               % Make sorted arc_list into
                                       % arc array.
    generate_constraints(Selection).

find_all_arc_lists(_) :-               % Find all arcs that are in the
                                       % arc-path incidence matrix.
```

```
    retrieveh(matrix,_,arc_path_matrix(AN,_)),
    int_text(AN, Arc_String),
    [! concat([$0000$, Arc_String, $,$], New_Arc_String),
       retract(init_string(Init)),
       ( string_search(New_Arc_String,Init,_), % Do not include the
         asserta(init_string(Init))            % duplicate arc string.
       ;
         concat(New_Arc_String, Init, New_String), % Append the new
         asserta(init_string(New_String))          % string.
       )
    !],
    fail.


find_all_arc_lists(Final) :-
    retract(init_string(Main_String)),
    string_length(Main_String, Length),
    dec(Length,Pos),
    substring(Main_String,0,Pos,New_String),
    concat([$[$, New_String, $]$], Output_String),
    string_term(Output_String, Final), !.      % Change string into
                                               % a list.


/*-------------------------------------------------------------*/

make_arc_array([]).                  % Seperate the arc_list and
make_arc_array([Anbr|Rest]) :-       % put it into an array format.
    assertz(arc_array(Anbr)),
    make_arc_array(Rest).


/*-------------------------------------------------------------*/

generate_constraints(Selection) :-
    retract(arc_array(Anbr)),
    [! cap(Anbr,Capacity),
       generate_constraint_inequality(Anbr),
       write(' '),
       write('<= '),
       write(' '),
       ( Selection = 4,
         prob(Anbr,Pb),
         Expected_Cap is Capacity * Pb,
         write(Expected_Cap)
       ;
         write(Capacity)
       )
    !],
    fail.
```

```
generate_constraints(_).

generate_constraint_inequality(Anbr) :-
    nl, nl,
    write(' Arc '),
    write(Anbr),
    write(': '),
    ctr_set(10,0),        % First time flag; this is used to
                              % control when to print '+' in
                              % the constraint equation.
    ctr_set(4,1),        % Initialize counter four to 1;
                              % this counter keeps track of
                              % how many terms are printed in
                              % the constraint equation.
    output_paths_containing_Anbr(Anbr).


output_paths_containing_Anbr(Anbr) :-
    removeh(matrix,Anbr,arc_path_matrix(Anbr,Pnbr)),
    [! ( ctr_inc(10,Flag),
          Flag = 0              % If first term, don't print '+'.
        ;
          write(' + ')
        ),
        ( ctr_inc(4,Value),     % Get current term number and
                                % increment the counter four by 1
          Value > 7,
          Mod_Value is Value mod 7,
          Mod_Value = 1,
          nl, write('        ')
        ;
          true
        ),
        write('f'),
        write(Pnbr)
    !],
    fail.

output_paths_containing_Anbr(_).

/*******************************************************************/
/* 'lp_investment_formulations' generates formulations of invest- */
/* ment strategy mode 1 and 2 to improve the lower bound.        */
/*******************************************************************/


lp_investment_formulations(Selection,Where) :-
    ( Where = 1,
      get_investment_model(Selection)
    ;
      executing_message,
```

```
      ( Selection = 5,
        stdout('output5.lp',get_investment_model(Selection))
      ;
        stdout('output6.lp',get_investment_model(Selection))
      ),
      exit_popup,
      done_message(Selection)
    ).

get_investment_model(Selection) :-
    invest_model_title(Selection),
    invest_model_objective(Selection),
    invest_model_constraints(Selection),
    nl, nl,
    write('* ------ end ------ *'),
    nl,nl,nl,nl.

invest_model_title(Selection) :-
   write(' ..Title'),
   nl,nl,
   ( Selection = 5,
     write(' Investment Strategy Model 1')
   ;
     write(' Investment Strategy Model 2')
   ).

invest_model_objective(Selection) :-
    nl, nl,
    write(' ..Objective Maximize'),
    nl, nl,
    ctr_set(1,1),                       % path number
    ctr_set(3,1),                       % no. of terms in a line
    find_invest_model_objective(s,t,Selection).

find_invest_model_objective(Start,Goal,_) :-
    depth_first([Start],Goal,Path),
    find_reliability(Path,Rel),
    ctr_inc(1,Pnbr),
    output_invest_model_objective_variables(Pnbr,Rel),
    make_arc_path_matrix(Path,Pnbr),
    fail.

find_invest_model_objective(_,_,Selection) :-
    asserta(init_invest_string($$)),
    get_investment_variables(Invest_Vars),
    sort(Invest_Vars,Sorted_Vars),
    make_cap_array(Sorted_Vars),
    nl,
    ctr_set(2,1),
    ctr_set(10,0),                          % First time flag.
    ( Selection = 5
```

```
    ;
      write(' [ ')
    ),
    output_investment_variables(Selection),
    ( Selection = 5
    ;
      write(' ] ')
    ).

find_invest_model_objective(_,_,_).

make_cap_array([]).
make_cap_array([Anbr|Rest]) :-
    assertz(cap_array(Anbr)),
    make_cap_array(Rest).


output_invest_model_objective_variables(Pnbr,Rel) :-
    write(' '),
    write(Rel),
    write(' f'),
    write(Pnbr),
    write(' +'),
    ( ctr_inc(3,Value),
      Mod_Value is Value mod 4,
      Mod_Value = 0,
      nl
    ;
      true
    ), !.


get_investment_variables(_) :-
    cap(AN,Capacity),
    [! ( not number(Capacity)
       ;
         int_text(AN, Arc_String),
         concat([$0000$, Arc_String, $,$], New_Arc_String),
         retract(init_invest_string(Init)),
         concat(New_Arc_String, Init, New_String),
         asserta(init_invest_string(New_String))
)
    !],
    fail.


get_investment_variables(Final) :-
    retract(init_invest_string(Main_String)),
    string_length(Main_String, Length),
    dec(Length,Pos),
    substring(Main_String,0,Pos,New_String),
```

```
    concat([$[$, New_String, $]$], Output_String),
    string_term(Output_String, Final), !.


output_investment_variables(Selection) :-

   cap_array(Anbr),
   [!  ( ctr_inc(10,Flag),
         Flag = 0
       ;
         write(' +')
       ),
       ( ctr_inc(2,Value),
         Value > 7,
         Mod_Value is Value mod 7,
         Mod_Value = 1,
         nl
       ;
         true
       ),
       ( Selection = 5,
         write(' 0 d')
       ;
         write(' 0 g')
       )
   !],
   write(Anbr),
   fail.

output_investment_variables(_).

/*-----------------------------------------------------*/


invest_model_constraints(Selection) :-

    nl, nl,
    write(' ..Constraints'),
    asserta(init_string($$)),
    find_all_arc_lists(Arc_List),
    sort(Arc_List,Sor),
    make_arc_array(Sor),
    generate_arc_constraints(Selection),
    ctr_set(10,0),                % counter for budget term
    nl, nl,
    write(' Budget: '),
    generate_budget_constraint(Selection).


generate_arc_constraints(Selection) :-
    retract(arc_array(Anbr)),
```

```
    [! cap(Anbr,Capacity),
       generate_constraint_inequality(Anbr),
       ( Selection = 5,
         write(' - d')
       ;
         write(' - '),
         invest(Anbr,Amount),
         write(Amount),
         write(' g')
       ),
       write(Anbr),
       write(' '),
       write('<= '),
       write(' '),
       write(Capacity)
    !],
    fail.

generate_arc_constraints(_).


generate_budget_constraint(Selection) :-
   retract(cap_array(Anbr)),
   [! cost(Anbr,Unit_cost),
      ( Selection = 5,
        Cost is Unit_cost
      ;
        invest(Anbr,Amount),
        Cost is Unit_cost * Amount
      ),
      print_budget_terms(Anbr,Cost,Selection)
   !],
    fail.

generate_budget_constraint(_) :-
    ctr_set(5,1),
    budget(Budget),
    write(' '),
    write(' <= '),
    write(Budget).


print_budget_terms(Anbr,Cost,Selection) :-
   ( ctr_inc(10,Flag),
     Flag = 0
   ;
     write(' + ')
   ),
   ( ctr_inc(5,Value),
     Value > 5,
     Mod_Value is Value mod 5,
```

```
      Mod_Value = 1,
      nl, write('        ')
    ;
      true
    ).
    write(Cost),
    ( Selection = 5,
      write(' d')
    ;
      write(' g')
    ),
    write(Anbr).

/************************************************************/
/* 'nlp_investment_formulation' generates formulation of    */
/* investment strategy model 3.                             */
/************************************************************/

 nlp_investment_formulation(Selection,Where) :-

     ( Where = 1,
       begin_model_building
       ;
       executing_message,
       stdout('output7.nlp',begin_model_building),
       exit_popup,
       done_message(Selection)
     ).

 begin_model_building :-
     nl, nl, nl,
     write('MODEL: '), nl,
     invest3_objective, nl,
     invest3_constraints,
     write('LEAVE'),
     nl,nl,nl.

/*****************************************************************/
/*-- Objective function  --*/

 invest3_objective :-
     write('MAX= '),
     ctr_set(1,1),  % Counter one generates path number.
     ctr_set(3,1),  % Initialize counter three to 1
                         % to keep track of how many terms
                         % are printed in a line in the objective
                         % function.
     find_invest3_objective(s,t),
     write(' ;').
```

```
find_invest3_objective(Start,Goal) :-
   depth_first([Start],Goal,Path),
   ctr_inc(1,PathNmbr),                 % Get current path number and
                                     % increment the counter one by one.
   print_invest3_objective(PathNmbr),
   make_arc_path_matrix(Path,PathNmbr),  % Make arc-path incidence
                                      % matrix.
   fail.

find_invest3_objective(_,_).

/*---------------------------------------------------------------*/

print_invest3_objective(PathNmbr) :-
   ( PathNmbr = 1
   ;
     write(' + ')
   ),
   ( ctr_inc(3,Value),    % Get current variable number and
                          % increment the counter three by 1.
     Value > 5,
     Mod_Value is Value mod 5,   % If the remainder of Value
     Mod_Value = 1,              % divided by 4 is 1, then skip
     nl,                         % to next line.
     write('     ')
   ;
     true
   ),
   write('R'),
   write(PathNmbr),
   write(' * F'),                % Print path flow variable.
   write(PathNmbr), !.

/****************************************************************/
/* constraint functions */


invest3_constraints :-                 % Get constraints function.
    asserta(init_string($$)),          % Initialize to empty string.
    find_all_arc_lists(Arc_List),
    sort(Arc_List,Sor),                % Sort Arc_List in ascending order.
    make_arc_array(Sor),               % Make sorted arc_list into
                                       % arc array.
    ctr_set(1,1),            % Counter 1 contains pathnumber.
    generate_rj_descriptions,
    generate_arc_prob_constraints,
    ctr_set(9,0),                     % Flag counter to control line feed.
    generate_path_flow_constraints,
    ctr_set(10,0),            % Flag counter to control + sign.
    generate_rel_budget_constraints,
    write('END'), nl,
```

```
    generate_x_nonnegativity,
    ctr_set(1,1),           % Path number counter
    generate_f_nonnegativity.

/*------------------------------------------------------------------*/

 generate_rj_descriptions :-
   depth_first([s],t,Path),
   [! ctr_inc(1,PathNmbr),
      write('R'),
      write(PathNmbr),
      write(' = '),
      sort(Path,SortedPath),
      ctr_set(7,1),     % Counter to control the number of terms.
      write_rhs_of_equality(SortedPath),
      write(';'),
      nl
   !],
   fail.

 generate_rj_descrip' .ons.

 write_rhs_of_equality([]).

 write_rhs_of_equality([Arc|Rest]) :-
     ( not number(Arc)
     ;
       ( ctr_inc(7,Value),
         Value > 3,
         Mod_Value is Value mod 3,
         Mod_Value = 1,
         nl, write('     ')
       ;
         true
       ),
       write('( '),
       prob(Arc,Probability),
       write(Probability),
       write(' + .1 * X'),
       write(Arc),
       write(' ) '),
       ( Rest == [s,t]
         ;
         write('* ')
       )
     ),
     write_rhs_of_equality(Rest).

/*--------------------------------------------------------------*/

 generate_arc_prob_constraints :-
```

```
   prob(ArcNmbr,Probability),
   [! ( not number(ArcNmbr)
      ;
        write(Probability),
        write(' + .1 * X'),
        write(ArcNmbr),
        write(' < 1 ;'),
        nl
      )
  !],
  fail.

generate_arc_prob_constraints.

/*-----------------------------------------------------------------*/

generate_path_flow_constraints :-
    retract(arc_array(ArcNmbr)),
    [! cap(ArcNmbr,Capacity),
       generate_path_flow_constraint_inequality(ArcNmbr),
       write(' '),
       write('< '),
       write(Capacity),
       write(' ;')
    !],
    fail.

generate_path_flow_constraints :- nl.

generate_path_flow_constraint_inequality(ArcNmbr) :-
    ( ctr_inc(9,Flag),
      Flag = 0
    ;
      nl
    ),
    ctr_set(10,0),       % First time flag; this is used to
                              % control when to print '+' in
                              % the constraint equation.
    ctr_set(4,1),       % Initialize counter four to 1;
                              % this counter keeps track of
                              % how many terms are printed in
                              % the constraint equation.
    output_paths_containing_ArcNmbr(ArcNmbr).


output_paths_containing_ArcNmbr(ArcNmbr) :-
    removeh(matrix,ArcNmbr,arc_path_matrix(ArcNmbr,PathNmbr)),
    [!  ( ctr_inc(10,Flag),
          Flag = 0            % If first term, don't print '+'.
        ;
          write(' + ')
```

```
        ),
        ( ctr_inc(4,Value),     % Get current term number and
                                % increment the counter four by 1
          Value > 9,
          Mod_Value is Value mod 9,
          Mod_Value = 1,
          nl
        ;
          true
        ),
        write('F'),
        write(PathNmbr)
    !],
    fail.

output_paths_containing_ArcNmbr(_).

/*-----------------------------------------------------------------*/

generate_rel_budget_constraints :-    % probably can revise
                                        % this: I dont think there
                                        % is any need for using
                                        % prob-array from
                                        % get-invest3-variables
   prob(ArcNmbr,_),
   [! ( not number(ArcNmbr)
      ;
       rcost(ArcNmbr,Unit_cost),
       print_rel_budget_terms(ArcNmbr,Unit_cost)
      )
   !],
   fail.

generate_rel_budget_constraints :-
    ctr_set(5,1),
    rbudget(Budget),
    write(' '),
    write(' < '),
    write(Budget),
    write(' ;'),
    nl.


print_rel_budget_terms(ArcNmbr,Cost) :-
   ( ctr_inc(10,Flag),
     Flag = 0
   ;
     write(' + ')
   ),
   ( ctr_inc(5,Value),
     Value > 5,
```

```
      Mod_Value is Value mod 5,
      Mod_Value = 1,
      nl
    ;
      true
    ),
    write(Cost),
    write(' * X'),
    write(ArcNmbr).

/*-------------------------------------------------------------*/

generate_x_nonnegativity :-
   prob(ArcNmbr,Probability),
   [! ( not number(ArcNmbr)
      ;
       write('SLB X'),
        write(ArcNmbr),
        write(' 0 '),
        nl
      )
   !],
   fail.

generate_x_nonnegativity.

/*-------------------------------------------------------------*/

 generate_f_nonnegativity :-
    depth_first([s],t,_),
    [! ctr_inc(1,PathNmbr),
       write('SLB F'),
       write(PathNmbr),
       write(' 0 '),
       nl
    !],
    fail.

 generate_f_nonnegativity.


/*-------------------------------------------------------------*/

 :- reconsult('windows2.ari').

/* ----------------------- end ---------------------------- */
```

```
/*****************************************************************/
/*                                                               */
/*                      WINDOWS2.ARI                             */
/*                                                               */
/*  This file contains windows to communicate with the user.    */
/*                                                               */
/*****************************************************************/

/*---------------------------------------------------------------*/
/*  Introduction Screen                                          */
/*---------------------------------------------------------------*/

windows :-
  cls,
  define_window(program_title,'',(23,0),(23,79),(91,0)),
  define_window(intro,'',(0,0),(22,79),(26,0)),
  current_window(_,program_title),
  tmove(0,12),
  write(' FORMULA  Version 2.0      AFIT  October, 1990'),
  current_window(_,intro),
  define_intro_window.

 define_intro_window :-
  nl,nl,
  tab(17),
  write('****************************************'),
  nl,tab(17),
  write('*    F O R M U L A   V e r  2.0    *'),
  nl,tab(17),
  write('****************************************'),
  nl,nl,nl,
  tab(11),
  write('This program finds all paths in the network from '),
  nl,tab(11),
  write('source to sink and calculates all paths reliabilities.'),
  nl,tab(11),
  write('It also generates six mathematical programming '),
  nl,tab(11),
  write('models that will assist in analyzing the performance '),
  nl,tab(11),
  write('of the network and in determining the investment '),
  nl,tab(11),
  write('strategy to improve the performance of the network.'),
  nl,tab(11),
  write('These models are developed based on the arc-path '),
  nl,tab(11),
  write('incidence matrix built from the description of the '),
  nl,tab(11),
  write('network in the input file. '),
  nl,nl,tab(11),
  write('PLEASE make sure the input file contains correct'),
```

```
  nl,tab(11),
  write('description of the network to be analyzed.'),
  nl,nl,tab(23),
  write('Press any key to continue. '),
  get0(_),
  cls.


/*-------------------------------------------------------------*/
/* Asks for the input file name. If the file name is not found, */
/* the program prints the error message; otherwise, consults    */
/* the input file.                                              */
/*-------------------------------------------------------------*/

open_input_datafile :-
  create_popup(query1,(7,20),(14,60),(62,-62)),
  write(' Please type in your input file name.   '),
  tmove(3,2),
  write(' > '),
  read_line(0,File),
  (
    consult_file(File),
    exit_popup
  ;
    display_filename_error,
    exit_popup,
    open_input_datafile
  ).

consult_file(File) :-
   stdin(File,_),
   consult(File).

display_filename_error :-
  create_popup(error1,(16,20),(21,60),(79,-79)),
  write('     Error: File not found. '),
  put(7),
  nl, nl,
  write(' Type in any key to continue or '),
  nl,
  write(' press RETURN to exit. '),
  get0(Reply),
  ( Reply = 13,
    exit_popup,
    exit_popup,
    clear_windows
  ;
    exit_popup
  ).
```

```
/*--------------------------------------------------------------*/
/* Ask user what to do.                                         */
/*--------------------------------------------------------------*/

get_selection_number(Selection) :-
  create_popup(query2,(3,12),(19,68),(62,-62)),
  tmove(1,16),
  write('How may I help you?'),
  tmove(4,2),
  write('1. Find all paths and calculate path reliabilities.'),
  tmove(5,2),
  write('2. Generate the Maximum Flow Formulation.'),
  tmove(6,2),
  write('3. Generate the Lower Bound Formulation. '),
  tmove(7,2),
  write('4. Generate the Upper Bound Formulation. '),
  tmove(8,2),
  write('5. Generate the Investment Strategy Model 1. '),
  tmove(9,2),
  write('6. Generate the Investment Strategy Model 2. '),
  tmove(10,2),
  write('7. Generate the Investment Strategy Model 3. '),
  tmove(11,2),
  write('8. Exit'),
  tmove(13,18),
  write('Type in number >  '),
  get0(Choice),            % The selection chosen is in ASCII code,
  exit_popup,              % that is one is represented as 49, two is
  Sel_Nbr is Choice - 48,  % represented as 50, etc. Thus, 48 is
  ( Sel_Nbr >= 1,          % substracted to make it back to regular
    Sel_Nbr =< 8,          % arabic number.
    Selection = Sel_Nbr
  ;
    put(7),
    get_selection_number(Selection)
  ).


/*--------------------------------------------------------------*/
/* Asks user where to display the output.                       */
/*--------------------------------------------------------------*/

get_where_to_send_output(Where) :-
  create_popup(query3,(5,20),(16,60),(62,-62)),
  tmove(1,8),
  write('Where do you want the'),
  tmove(2,8),
  write('output displayed ?'),
  tmove(4,11),
  write('1. Screen'),
  tmove(6,11),
```

```
  write('2. File'),
  tmove(8,8),
  write('Type in Number >  '),
  get0(Choice),
  exit_popup,
  Sel_Nbr is Choice - 48,
  ( Sel_Nbr >= 1,
    Sel_Nbr =< 2,
    Where = Sel_Nbr
  ;
    put(7),
    get_where_to_send_output(Where)
  ).


/*--------------------------------------------------------------*/
/* Asks user to run the program again with same input file.     */
/*--------------------------------------------------------------*/

get_run_again_reply(Reply):-
  create_popup(query4,(20,0),(22,79),(62,-62)),
  write(' Do you want to run the program again (y or n) ?  '),
  get0(User_Reply),
  exit_popup,
  (
    ( User_Reply = 121;      % y
      User_Reply = 110      % n
    ),
    Reply = User_Reply
  ;
    put(7),
    get_run_again_reply(Reply)
  ).


/*--------------------------------------------------------------*/
/* Prints 'executing' message while output is sent to an output */
/* file.                                                        */
/*--------------------------------------------------------------*/

executing_message :-
  create_popup('',(11,25),(14,50),(207,79)),
  write('      Executing ... '),
  nl,
  write('      Please Wait.').


/*--------------------------------------------------------------*/
/* Prints 'Done' message after output is sent to an output file. */
/*--------------------------------------------------------------*/
```

```
done_message(Output_File) :-
  create_popup('',(11,20),(14,57),(58,58)),
  write('              Done. '),
  nl,
  write('  Output was sent to "output'),
  write(Output_File),
  ( Output_File = 7,
    write('.nlp".')
  ;
    write('.lp".')
  ).


/*-----------------------------------------------------------------*/
/* Clears all windows before exiting(quiting) the program.        */
/*-----------------------------------------------------------------*/

clear_windows :-
  delete_window(program_title),
  delete_window(intro),
  abolish(arc/2),
  abolish(prob/2),
  abolish(cap/2),
  abolish(cost/2),
  abolish(budget/1),
  removeallh(matrix),
  current_window(_,main).


/*----------------------------- end ------------------------------*/
```

Appendix D. *Application of Models in a Space Environment*

The proliferation of space-based communication networks has steadily increased over the past 30 years. Because of this, the United States has become more dependent on space-based communication networks. It was for this reason that a discussion was included on improving the throughput of stochastic communication networks that have some of their components based in space. The discussion does not analyze a specific network, but rather shows the reader how to apply the methodology described in Chapter IV to a space-based network.

A space-based stochastic communication network has both terrestrial and extraterrestrial components. Figure 19 shows a picture of a typical space-based communications network. As seen in the figure, the network consists of ground stations, denoted by radar antennas, and communication satellites. The satellites are generally in a geostationary orbit above the equator. A geostationary orbit means the orbital period of the satellite is the same as that of the earth, i.e., 24 hours. Because of this, the satellite will always appear in the same position relative to a specific ground station. The communication links shown in Figure 19 represent modulated signals in the RF (Radio Frequency) spectrum that are transmitted through the earth's atmosphere. These signals can go in either direction, from the satellites to the ground stations or vice versa. In addition to the ground station to satellite communications links, there are some microwave links between ground stations. After looking at Figure 19, the reader might get the impression that a space-based communication network is not much different than a terrestrial-based system. This is not true! While the two types of networks may appear the be the same conceptually, they operate in entirely different environments. The space environment is much harsher than that of the earth. For this reason, a brief description of the space environment is discussed next.

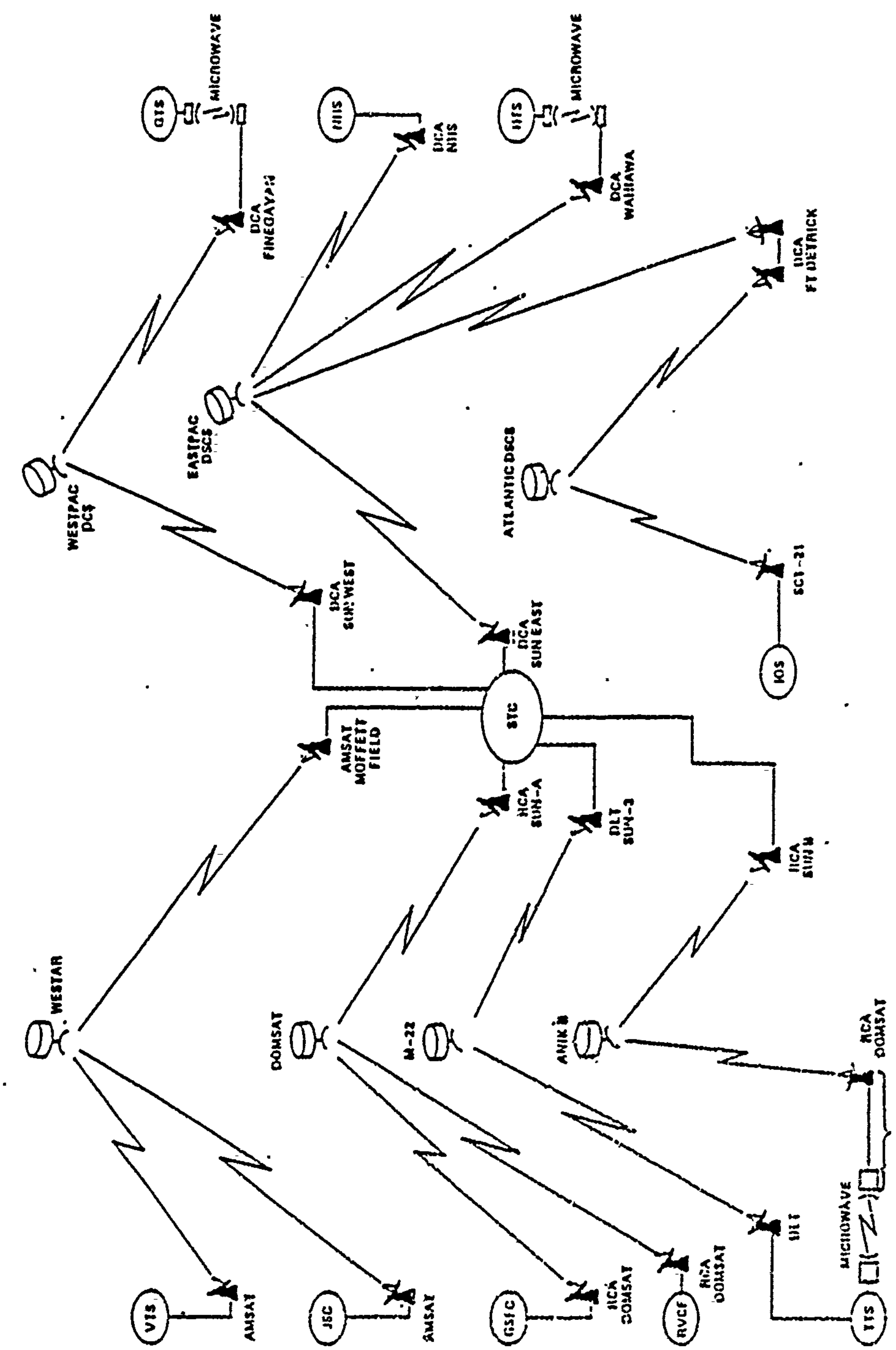


Figure 19. Typical Space Based Communications Network

D.1 The Space Environment

The primary components of the space environment are:

1. Gravitational Fields.
2. Vacuum.
3. Thermal Radiation.
4. Magnetic Fields.
5. Meteorites and Particles.

D.1.1 Gravitational Fields. There are three primary components to the gravity forces that effect earth orbiting satellites: the earth's, moon's, and sun's gravitational fields. The earth's gravitational field is not uniformly spherical due to the non-homogeneous distribution of the mass of the earth. The sun and moon both apply gravitational forces on orbiting satellites.

D.1.2 Vacuum. Space is dominated by a vacuum. As altitude increases, the density of air molecules decreases. This decreases the atmospheric drag on the orbiting satellite, and eventually the drag will become negligible.

D.1.3 Thermal Radiation. There are two types of thermal radiation that impact on a satellite: solar and terrestrial. The solar radiation flux present on a geostationary satellite is approximately 1353 watts per square meter. In addition, the satellite is being radiated by about 40 watts per square meter from the earth. When the satellite is in view of the sun it feels both sources of radiation, but only senses the earth's radiation when in the earth's shadow.

D.1.4 Magnetic Fields. The earth's magnetic field is a dipole that forms an angle of 11.5 degrees with respect to the rotational axis. The magnetic field creates

a torque on the satellite and a magnetic moment. The global magnetic moment of a geostationary satellite is about $3.5x10^7$ N·m.

D.1.5 Meteorites and Particles. The earth is surrounded by a cloud of particles that the satellite must fly through. The velocities of the meteorites (material fragments, rocks, debris, etc.) vary from a few km/sec to several tens of km/sec. The kinetic energy of such meteorites is quite large. In addition to meteorites, the satellite is continuously bombarded by cosmic particles. Even though these particles are very small when compared to meteorites, they travel at phenomenal speeds. For example, the solar winds, which consists mainly of protons and electrons, travel at about 400 km/sec (16:183-189).

D.2 Effects of the Space Environment on Satellites

The space environment has the following effects on an orbiting satellite:

1. Mechanical perturbations as a result of forces, such as gravity and torques, effect a satellite's attitude and orbit.
2. The thermal effects are quite dramatic. A satellite heats up when it's in the view of the sun, but cools down when it falls in the earth's shadow.
3. The impacting of high energy radiation and particles degrades the satellite's materials and surface finish.

As a result, the satellites are not fully reliable, and are therefore stochastic (16:183).

D.3 Launching and Transfer Orbits

Before a satellite achieves a geostationary orbit it must be launched and placed into a transfer orbit. The launch is quite stressful on the satellite. It must endure shocks and vibrations throughput the launch, from the launch pad to the injection into the transfer orbit. The transfer orbit is an elliptical orbit (200 to 36,000 km) that

takes the satellite from a low earth orbit to the higher geostationary orbit. During the transfer orbit, the satellite is effected by atmospheric drag and the earth's thermal and magnetic properties (16:201).

D.4 Improving Reliability in Satellites

Once a satellite is launched there is very little that can be done to improve the reliability. Since it is impossible to have "hands on" the satellite, the only way to make the satellite perform better is by uploading commands to change the onboard computer programs. For this reason, the reliability must be improved before the satellite is launched. By spending more money to increase the reliability of an individual satellite, it is possible to make the overall network more reliable and improve the throughput. Each satellite can be thought of as a node in the network. The following sections describe options available to the engineer for improving reliability in a satellite.

D.4.1 Preventing Infant Mortality Failures. A large number of parts fail immediately after they are placed into service. To prevent this from happening an extensive screening test must be used. By placing the parts through a series of vibration and X-ray tests, it is possible to identify the parts that are likely to fail early. This type of testing can increase the cost of the part by 10 to 20 times the original price.

D.4.2 Providing Reliability Through Quality Control. A foreign particle such as dust or grit can cause a component to fail once it is placed under the stress of operating in space. By the use of filters and "clean" rooms, it is possible to eliminate a large majority of these impurities. The manufacturing environment will resemble a hospital operating room. Inspections is another way to increase reliability through quality control. Training can also make employees aware of the needs for quality control. Improved quality control adds costs to the manufacturing process.

D.4.3 Providing Reliability Through Testing. The testing of subsystems to ensure they work both individually, and as part of the total system will increase reliability. Better yet, the use of thermal-vacuum rooms can simulate the environment that the system must operate in once it is deployed in space.

D.4.4 Testing of One-shot Devices. Many parts are designed to be used only once, such as rocket engines. Testing of these types of parts is destructive, and therefore costly. By testing a small sample, it is possible using statistics to estimate how good the entire batch is. Obviously, the more tests that are done, the better the statistics, and increases the chances of identifying batches which inferior parts. The elimination of inferior parts increases the reliability of the system.

D.4.5 Use of Redundancy. The use of redundant parts will show drastic improvements in reliability. However, it is more expensive to design and manufacture redundant systems. In addition, redundant systems add extra weight to the satellite creating the need for larger launch vehicles (1:10.7-10.11).

D.5 Improving Reliability in Satellite Communication Links

There are several factors that govern the design of geostationary communication satellites. They are:

1. The weight of the satellite.
2. The DC power that can be generated by the satellite's power supplies.
3. The available frequency bands.
4. The maximum dimensions of the ground station and satellite antennas.
5. The multiplexing technique used to allow a satellite to be shared by numerous ground stations.

Communication links have three primary components: transmitter, receiver, and transmission medium. The transmitters and receivers can be made more reliable through the use of high gain antennas and low signal to noise subsystems. The transmission medium is the earth's atmosphere, and this cannot be modified. The use of specified frequencies (the 4 and 6 GHz bands are primarily used by satellite communication systems) will increase the probability of the transmission getting through (18:104-105).

D.6 Improving Reliability in Ground Stations.

The ground station may be located on land, a ship, or an aircraft. Wherever the ground station is located, it must have a line of sight to the geostationary satellite. The best way to improve the transmission capabilities of the ground station is to achieve a low system noise temperature in the receiving channel. Doing so will increase the carrier-to-noise ratio, C/N, in the receiving channel. Since the gain-to-temperature ratio, G/T, is directly proportional to C/N, it is a useful parameter to characterize ground stations. The gain of the receiver, G, can be increased by making the antenna aperture size larger, but this is extremely expensive. The other option is to lower the temperature. The ideal case would be to lower the temperature toward zero, but this is not practical. Therefore a compromise must be made between increasing the size of the antenna and lowering the temperature (18:353-354).

D.7 Converting the Physical Network to a Graph

The network depicted in Figure 19 must be converted into a graph consisting of nodes and arcs. This process consists of drawing a diagram where the ground stations and satellites are nodes and the communication links are arcs. Once the nodes and arcs have been drawn then the next step is to compute the capacities and reliabilities of each node and arc. This information must come from the engineers that design the components. The engineers must assess the impacts of the space environment on

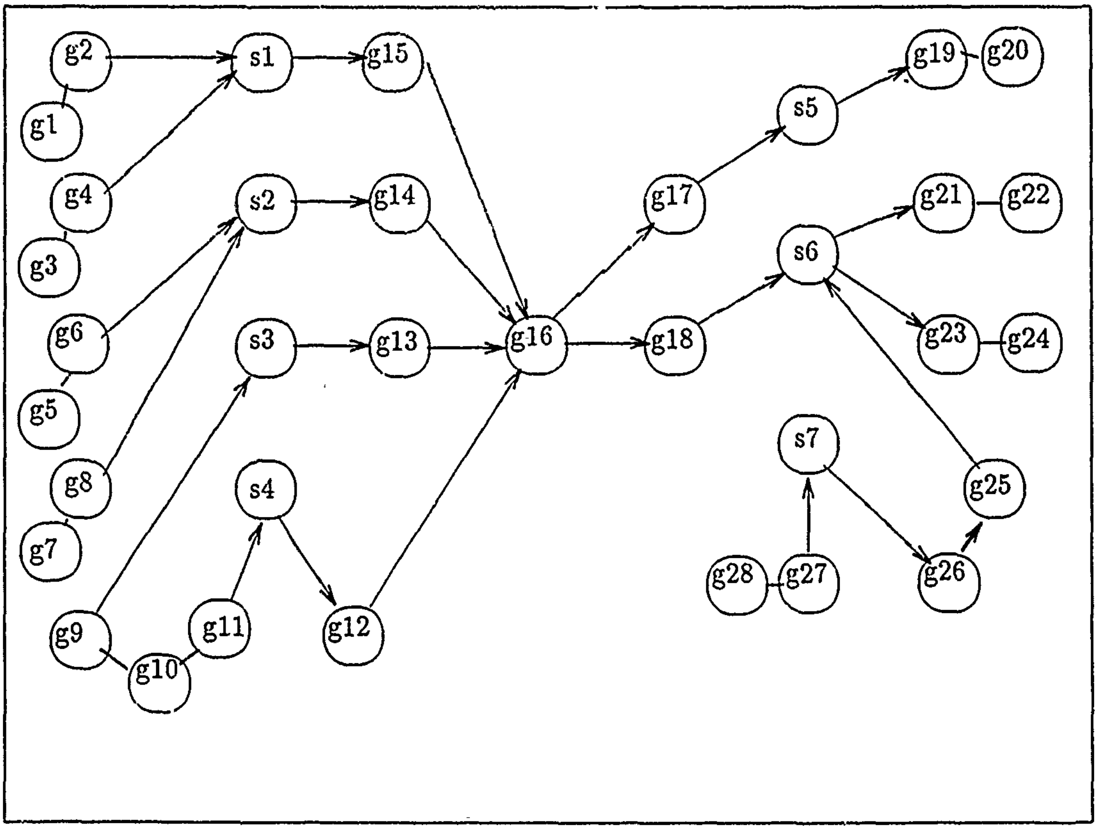


Figure 20. Graph of a Space Based Network

Table 32. Satellite Nodes

Node Number	Satellite Type
s1	WESTAR
s2	DOMSAT
s3	M-22
s4	ANIK B
s5	WESTPAC DSCS
s6	EASTPAC DSCS
s7	Atlantic DSCS

the reliability of the individual transmitters, receivers, power supplies, antennas, etc. to determine the reliability of the subsystems that make up the arcs and nodes. This in-depth assessment will result in a reliability value between 0 and 1 for a specific arc or node, as well as it's capacity. The capacity will be specified as transmission rate, such as a million bytes per second. A packet switched network will have a capacity of packets per second. Figure 20 shows the network from Figure 19 after it has been converted to a graph. Tables 32 and 33 show how the graph in Figure 20 corresponds to the physical diagram in Figure 19. The reader should not be alarmed that he/she may not be familiar with the types of ground stations and satellites listed in the tables. The reason they are listed is to make the reader aware that a space-based communication network is made up of many different types of equipment, i.e., the nodes are not homogeneous.

D.8 Running the Experiment

After the network is converted into a graph, the next step is to run the experiment described in Chapter VII to determine the lower and upper bounds, expected throughput, expected reliability, and optimum investment strategies. This data will be used to assess the performance of the network, and determine where improve-

Table 33. Ground Station Nodes

Node Number	Station Type
g1	VTS
g2	AMSAT
g3	JSC
g4	AMSAT
g5	GSFC
g6	RCA DOMSAT
g7	RVCF
g8	RCA DOMSAT
g9	DLT
g10	TTS
g11	RCA DOMSAT
g12	RCA SUN B
g13	DLT SUN-3
g14	RCA SUN-A
g15	AMSAT Moffett Field
g16	STC
g17	DCA SUN WEST
g18	DCA SUN EAST
g19	DCA Finegayan
g20	GTS
g21	DCA NHS
g22	NHS
g23	DCA Wahiawa
g24	HTS
g25	DCA Ft Detrick
g26	DCA Ft Detrick
g27	SCT-21
g28	IOS

ments should be made. The budget available and the costs to increase capacity and reliability must be known before the optimum investment strategies can be computed. This information can be gotten from the contracting and comptroller offices. If the network is made up of many subnetworks then the person running the experiment will have to contact the program manager's office for each subnetwork to get the budget and costing information. The results from the experiment must then be analyzed to see if the network is performing as designed, and where it can be improved.

D.9 Analyzing the Results from the Experiment

The analyst must answer two questions; will the network do the job and how can the network be improved? If improvements in the form of increased capacities and reliabilities are in order, then the engineers must be notified. Once the engineers have been notified, then they must assess the impacts to the schedule that would result if the improvements are implemented. Some of the improvements may require major redesigns. The optimum investment strategies are based strictly on improvements costs and the budgets available, and do not consider the impacts to the implementation schedule. As stated earlier, the increases to capacity and reliability for the satellites must be done before the satellite is launched, therefore improvements to the satellites themselves will impact the launch rate and schedule.

D.10 Deciding on an Improvement Strategy

After the results from the experiment have been analyzed then the next step is to decide on a plan for implementing the optimum investment strategy. Obviously any improvements to the satellites must be made during the design, engineering, and manufacturing processes, but improvements to ground stations could wait until the network is operational. These tradeoffs have to be made by the program managers office. The plan may consist of many phases which occur over an extended time

period, in some cases years.

D.11 Summary

The application of the methodology described in Chapter IV can be used for space-based communication networks, but not without some careful planning. It is possible to increase the capacity and reliability of communication satellites during the design, engineering, and manufacturing processes. However, once the satellites are launched into their geostationary orbits there is very little that can be done to improve the satellite. Ground stations and communication links offer the only candidates for improvement once the network is operational.

Appendix E. *Branch and Bound Solution to Network 4*

This appendix contains the integer solution to Investment Strategy Model 3 for Network 4. The integer solution was calculated by using the Branch and Bound algorithm in conjunction with Investment Strategy Model 3. Figure 21 shows the branch and bound tree that resulted.

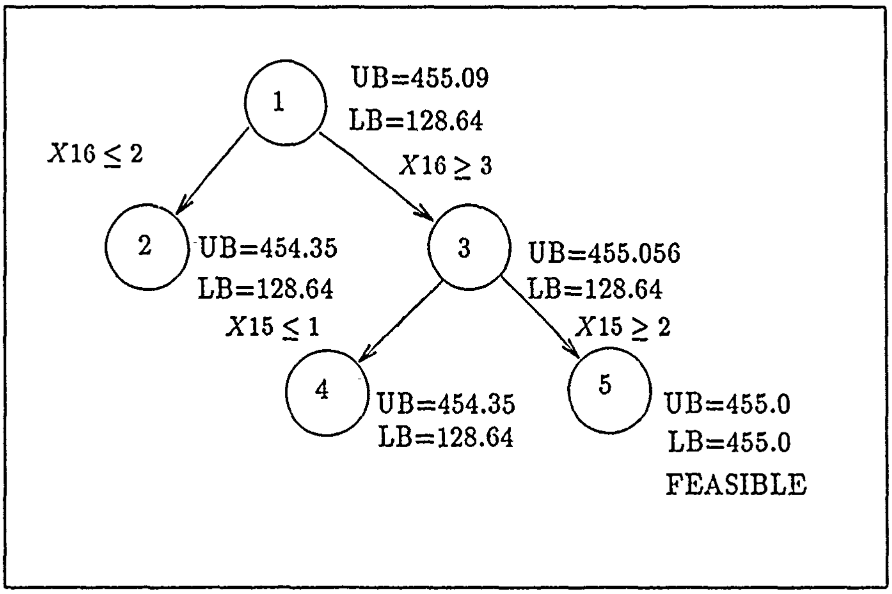


Figure 21. Branch and Bound Tree for Network 4

The following steps were used to arrive at the optimal solution:

1. First, Investment Strategy Model 3 was solved as strictly a nonlinear program as it is formulated in Chapter III. The resulting value of the objective function was 455.09 (see Node 1 in Figure 21). This is the same solution as reported in Table 14 of Chapter VIII. Therefore, 455.09 will be used as the upper bound. The lower bound will be 128.64, which is the solution to Investment Strategy Model 3 when the budget is set to zero. Table 34 shows the values of the X_i variables for the solution to Node 1.

2. The next step was to branch from Node 1. X16 was chosen as the branching variable since it was the farthest from being integral. Node 2 was the solution node when X16 was set to less than or equal to 2, and Node 3 as the solution when X16 was set to greater than or equal to 3. Node 2 had an optimal solution of 454.35, while Node 3 had 455.056. Tables 35 and 36 show the X_i values for Nodes 2 and 3 respectively. Neither table shows an integer solution. Since Node 3 had the higher optimal solution, then a decision was made to branch from it instead of Node 2.

3. At Node 3, the branching variable was X15. X15 was set to less than or equal to 1 first, and this resulted in an optimal solution of 454.35 for Node 4. This solution was not integer. Table 37 has the variables for the solution at Node 4.

4. Next, X15 was set to greater than or equal to 2, and this resulted in an integer solution at Node 5 (see Table 38). Since the optimal solution at Node 5 was greater than any of the other noninteger solutions (except Node 1 of course), then the solution at Node 5 must be the optimal integer solution. As indicated by Figure 21, the solution at Node 5 was 455.0. Table 38 shows the values of the X_i variables for Node 5 and they are all integer. Tables 39 and 40 show the X_i variables of the optimal solution, and the arcs and nodes they correspond to in Figure 8 of Chapter VI.

Table 34. Variables for Node 1

Variable	Value
X1	0
X2	0
X3	0
X4	0
X5	0
X6	0
X7	0
X8	0
X9	1.17
X10	0
X11	5.0
X12	0
X13	5.17
X14	4.0
X15	1.83
X16	2.83
X17	0
X18	0

Table 35. Variables for Node 2

Variable	Value
X1	0
X2	0
X3	0
X4	0
X5	0
X6	0
X7	0
X8	0
X9	1.5
X10	0
X11	5.0
X12	0
X13	5.5
X14	4.0
X15	2.0
X16	2.0
X17	0
X18	0

Table 36. Variables for Node 3

Variable	Value
X1	0
X2	0
X3	0
X4	0
X5	0
X6	0
X7	0
X8	0
X9	1.11
X10	0
X11	5.0
X12	0
X13	5.11
X14	4.0
X15	1.78
X16	3.0
X17	0
X18	0

Table 37. Variables for Node 4

Variable	Value
X1	0
X2	0
X3	0
X4	0
X5	0
X6	0
X7	0
X8	0
X9	1.50
X10	0
X11	5.0
X12	0
X13	5.50
X14	4.0
X15	1.0
X16	3.0
X17	0.0
X18	0.0

Table 38. Variables for Node 5

Variable	Value
X1	0
X2	0
X3	0
X4	0
X5	0
X6	0
X7	0
X8	0
X9	1.0
X10	0
X11	5.0
X12	0
X13	5.0
X14	4.0
X15	2.0
X16	3.0
X17	0
X18	0

Table 39. Integer Improvements for Arcs In Network 4

X_i Variable	Start Node	Terminate Node	Initial Reliability	Reliability Increase	Final Reliability
X13	3	5	0.300	0.500	0.800
X14	4	6	0.600	0.400	1.000
X16	5	6	0.700	0.300	1.000

Table 40. Integer Improvements For Nodes In Network 4

X_i Variable	Node Number	Initial Reliability	Reliability Increase	Final Reliability
X9	3	0.700	0.100	0.800
X11	4	0.500	0.500	1.000
X15	5	0.800	0.200	1.000

Bibliography

1. Air University. *Space Handbook.* AU-18. Air University Press, January 1985.

2. Aneja, Y. P. and K. P. K. Nair. "Maximal Expected Flow in a Network Subject to Arc Failures," *Network, 10* : 45-57 (1980).

3. Arity Corporation, *The Arity/Prolog Language Reference Manual.* Concord, Massachusetts, 1988.

4. Bailey, Capt Thomas G. *Response Surface Analysis of Stochastic Network Performance.* MS thesis, AFIT/GOR/ENS/88D-01. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH,December 1988 (AD-A202561).

5. Brecht, Timothy B. and Charles J. Colbourn. "Multiplicative Improvements in Network Reliability Bounds," *Network, 19*: 521-529 (1989).

6. Carey, Malachy and Chril Hendrickson. "Bounds on Expected Performance of Networks with Links Subject to Failure," *Network, 14*: 439-456 (1984).

7. Chan, Yupo, Deputy Department Head of Operational Sciences. Personal interview. Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 17 April 1990.

8. Chiou, Shen-Neng and Victor O. K. Li. "Reliability Analysis of a Communication Network with Multimode Components," *IEEE Journal on Selected Areas in Communications, Vol. SAC-4, No. 7*: 1156-1161 (October 1986).

9. De Neufville, Richard. *Applied Systems Analysis, Engineering Planning and Technology Management.* New York: McGraw Hill Publishing Company, 1990.

10. El-Sheikh, Talatt S. and Grant A. Cheston. "Recursive Evaluation of Overall Network Reliability," *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing, 16th.* 278-286. Monticello, IL, 1978.

11. Jensen, Paul A. *Microsolve/Operations Research IBM Revised Edition.* Oakland, CA: Holden-Day, 1985.

12. Kubat, Peter. "Assessing Throughput and Reliability in Communication and Computer Networks," *IEEE Transactions on Reliability, Vol. 37, No. 3*: 308-311 (August 1988).

13. Lam, Y. F. and Victor O. K. Li. "Reliability Modeling and Analysis of Communication Networks," *IEEE Transactions on Communications, Vol. COM-34, No. 1*: 82-84 (January 1988).

14. Li, Victor O. K. and John A. Silvester. "Performance Analysis of Networks with Unreliable Components," *Integrating Communication for World Progress; International Conference on Communications, Boston MA, conference record vol. 1.* 493-498. New York: IEEE, (June 1983).

15. Liebman, Judith and others. *Modeling and Optimization with GINO.* Palo Alto, CA: The Scientific Press, 1986.

16. Maral, G. and M. Bousquet. *Satellite Communication Systems.* New York: John Wiley and Sons, 1986.

17. Minoux, M. "Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications," *Network, 19*: 313-360 (1989).

18. Pratt, Timothy and Charles W. Bostian. *Satellite Communications.* New York: John Wiley and Sons, 1986.

19. Sancho, N. G. F. "A Functional Equation For Finding the Largest Expected Capacity of a Graph," *Network, 19*: 261-268 (1989).

20. SAS Institute Inc. *SAS/OR User's Guide, Version 6, 1st Edition.* Cary, NC, 1989.

21. Sunset Software. *LP/MIP-83: A Professional Linear and Mixed Integer Programming System.* San Marino, CA, 1985.

22. Yang, Che-Liang and Peter Kubat. "Efficient Computation of Most Probable States for Communication Networks with Multimode Components" *IEEE Transactions on Communications, Vol. 37*: 535-538 (May 1989).

23. Yim, Capt Eugene. *Improving the Survivability of a Stochastic Communication Network.* MS thesis, AFIT/GOR/ENS/88D-01. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH,December 1988 (AD-A202872).

24. —— and others. *Exact and Approximate Improvement to the Throughput of a Stochastic Network.* Draft report submitted to Joint Meeting of the Operations Research Society and the Institute of Management Science, Vancouver, B. C. Canada, May 1989.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1990

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Improving Reliability In A Stochastic Communication Network

5. FUNDING NUMBERS

6. AUTHOR(S)
William L. Gaught, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSO/ENS/90D-10

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This research investigated the behavior of increasing the reliability of components in stochastic communication networks. A stochastic communicaion network is one in which the components, links and nodes, are not 100 percent reliable. In addition to the reliability value, each component also has a capacity value that limits the flow through the component. The throughput in a stochastic communication network can be improved by increasing the capacity or the reliability of the individual components. A set of linear and nonlinear models was developed to determine the best investment strategy for increasing the reliability of individual components. Once the models were developed, an experiment was designed to compare the output of the models to another set of existing models that calculated the best investment strategy for increasing the capacity of individual components. The experiment employed a Prolog program to: enumerate the paths, compute the reliability of each path, and formulate the linear and nonlinear models for direct input into mathematical programming packages. In addition, the linear models were reformulated as networks with side constraints and solved using a network flow package. The models and methodology are general and could easily be modified. Keywords

14. SUBJECT TERMS
Networks, Reliability, Stochastic Networks, Communication Networks, Theses. (JHD)

15. NUMBER OF PAGES
313

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL